EDGE T-MATRIX IN NETWORK THEORY

by

Le Phung

United States
Naval Postgraduate School

THESIS

EDGE T-MATRIX IN NETWORK THEORY

by

Le Phung

September 1970

Edge T - Matrix in Network Theory

by

Le Phung
Lieutenant, Viet-Nam Navy
B.S., Mathematics, University of Saigon, 1958
B.S.E.E., Naval Postgraduate School, 1969

Submitted in partial fulfillment of the
quirements for the degree of

ELECTRICAL ENGINEER

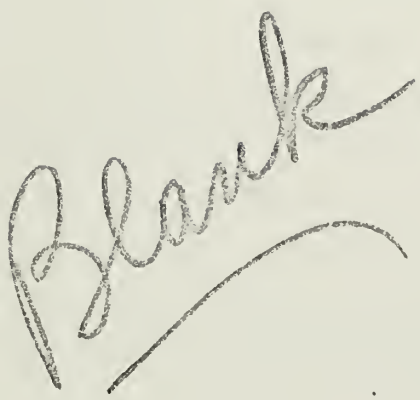
from the

NAVAL POSTGRADUATE SCHOOL
September 1970

ABSTRACT

Theory of the edge T-matrix will be developed and applied to derive algorithms for solving basic problems of network theory such as the determination of a fundamental loop or cut-set matrix, the path matrix, the circuit matrix, the seg matrix and the tree summation calculation. Application of the tree summation to the determination of sensitivity functions without actual derivative operation will also be investigated. Formulas for determining topologically the sensitivity function will be proved.

TABLE OF CONTENTS


Blank

LIST OF ILLUSTRATIONS

ACKNOWLEDGEMENT

The author wishes to thank his thesis advisor, Professor Shu-Gar Chan, for guidance during the period of research.


Blank

I. INTRODUCTION

Enumeration of subgraphs are interesting basic problems in graph theory. In network theory, the problems of finding (a) a fundamental loop or cutset matrix, (b) the circuit or seg matrix, (c) the path matrix, and (d) the set of trees in a graph are of primordial importance. Quite a number of papers have been written by various authors to solve these problems, particularly the problem of tree finding, which is the fundamental problem in topological analysis of electrical networks. Different theories, such as theory of groups, matrix theory, Boolean algebra, combinatorial analysis, have been applied for solving these problems. However, each of the existing methods suffers from one or more of the following disadvantages: (a) the method is too complicated for hand calculation or for computer implementation; (b) it often involves duplication; (c) large computer storage is required; and (d) the process requires long computation time.

In this thesis, the edge T-matrix will be defined and used as a new tool for representing a network. Based upon the properties of edge T-matrices, five algorithms for solving the above mentioned problems will be developed. The proposed algorithms will prevent the generation of duplicates and at the same time minimize the computer storage and the computation time.

Because graphs have been used as mathematical model in several different fields, there is a considerable proliferation of terminology. To avoid possible confusions, Chapter II of this paper will be devoted to the definitions of fundamental terms which will be used in the subsequent chapters. Theoretical development concerning the edge T-matrix will be discussed in Chapter III. Algorithms for finding fundamental loop and cutset matrices, for determining the circuit, path, and seg matrices will be given in Chapter IV. Techniques for calculating 1-trees and 2-trees will be presented in Chapter V. Applications of the results obtained in Chapter V for determining network functions and and sensitivity functions will be investigated in Chapter VI. In this chapter a topological formula for determining the sensitivity functions, without actual derivative operation, will be given.

II. BASIC DEFINITIONS

A. INTRODUCTION

Because graphs have been used as mathematical models in several different fields, there is a considerable proliferation of terminology. Throughout the literature it is found that not only are various words employed for the same concept but, more confusing, the same word is used sometimes for different concepts. In view of this chaotic situation, this chapter is devoted to list a set of basic definitions of terms and concepts which are to be used in later discussions. Some of the definitions of other familiar terms will be listed in Appendix A. Properties derived from basics definitions will be assumed without proofs, since the proofs of these statements are well known and their inclusions will be beyond the scope of this paper.

In the following development only connected graphs with no self-loop are considered. The word graph may refer to either a non-oriented or an oriented graph if not otherwise indicated. The numbers of nodes and edges of the graph are denoted by n and e respectively. The nodes are always numbered from 1 to n. The edges can be designated by numbers from 1 to e, or by e distinct letters.

B. THE INCIDENCE MATRIX

Definition 2.1 The incidence matrix, denoted by A_a, of a graph G of no self loop, is a matrix of order n x b with

with each row identified by a vertex and each column by an edge, that is:

$$A_a = \left[a_{ij}\right] \; n \times b$$

where

$$a_{ij} = \begin{cases} 1, & \text{if edge } j \text{ is incident with node } i \text{ and oriented away from it,} \\ -1, & \text{if edge } j \text{ is incident with node } i \text{ and oriented towards it,} \\ 0, & \text{if edge } j \text{ is not incident with node } j. \end{cases}$$

In the case of non-oriented graphs, the sign of a_{ij} is ignored.

Definition 2.2 The reduced incidence matrix, A, is the matrix of order (n-1) x b obtained from A_a by deleting any row of A_a.

Property 2.1 Each column of A_a contains exactly one +1 and one -1.

This property results directly from Definition 2.1.

Property 2.2 The rank of the matrix A_a is n-1.

The proof of this property has been given by Kirchhoff in [16].

Property 2.3 The matrices A and A_a are totally unimodular, i.e. all their square submatrices have determinant either zero or +1 or -1.

This property has been proved by Veblen and Franklin in [31].

C. THE CIRCUIT MATRIX

Definition 2.3 A circuit is a connected subgraph having precisely two edges incident with each node (of the subgraph).

An arbitrary orientation of the circuit may be defined in the obvious way.

Definition 2.4 The circuit matrix, B_a, is the matrix of order c x b with each row identified by a circuit and each column by an edge such that

$$B_a = \left[b_{ij}\right]_{c \times b}$$

where c is the total number of circuits of the graph, and

$$b_{ij} = \begin{cases} +1, & \text{if the edge } j \text{ is included in circuit } C_i \text{ and having the same orientation with } C_i \\ -1, & \text{if the edge } j \text{ is included in circuit } C_i \text{ and oriented in the opposite direction to the orientation of } C_i \\ 0, & \text{if the edge } j \text{ is not included in } C_i \end{cases}$$

Property 2.4 Let B_a and A_a be the circuit matrix and the incidence matrix of a graph G. Then

$$A_a B_a^T = 0$$

The proof of Property 2.4 has been given by Veblen in [30].

Property 2.5 The rank of B_a is b - n + 1.

Property 2.5 has been proved by Reed and Seshu in [28].

D. TREES AND COTREES

Definition 2.5 A tree of a connected graph G is a connected subgraph that contains all the nodes of G but no circuits. The edges contained in a tree are called the branches of that tree.

Definition 2.6 If t_i denotes a tree of a graph G, then the complement subgraph t_i' of t_i is called the cotree defined with respect to t_i. The edges contained in t_i' are called the chords of t_i.

Several important properties of tree graph are listed in the following.

Property 2.6 Every tree has at least two terminal nodes (nodes at which only one edge is incident).

Property 2.7 All trees of a connected graph of n nodes contain exactly n-1 branches.

Property 2.8 A cotree of a connected graph contains b-n+1 edges.

Property 2.9 For a connected graph, an n-1 minor of the reduced matrix A is a nonsingular if and only if the n-1 columns of the minor correspond to branches of a tree.

Property 2.10 The number of trees, T of a graph is

$$T = \det A.A^T$$

Definition 2.7 A k-tree of a connected graph is a set of k unconnected and circuitless subgraphs, which together

include all of the nodes of the graph and formed by n-k edges.

Definition 2.8 Consider a graph G together with four distinct nodes a, b, c and r. A type-I 2-tree, denoted by t(a;r) is a 2-tree of which node a is required to be in one of its subgraphs and node r in the other. A type-II 2-tree, denoted by t(a,b;r), is a 2-tree in which nodes a and b are required to be in one of its subgraphs and node r in the other. A type-III 2-tree is a 2-tree, designated by t(a,b;c,r), of which node a and b are required to be in one of its subgraphs and nodes c and r in the other. A type-IV 2-tree, denoted by t(a,b,c;r), is a 2-tree of which nodes a,b and c are required to be in one of its subgraphs and and node r in the other.

Definition 2.9 Given a directed graph G'. Let G be the graph derived from G' by ignoring the edge orientation. A subgraph t(r) of G' is called a directed tree if and only if:

(1) Ignoring the edge orientation, t(r) is a tree of G.

(2) From every node, except node r, called the reference node, there is exactly one outgoing directed edge. No edge of t(r) is directed away from the reference node r.

Definition 2.10 Given a directed graph G' together with four nodes a,b,c and r. Let G be the graph derived from G' by ignoring the edge orientation. A type-I directed 2-tree, denoted by t'(a;r) is a 2-tree defined from G' if ignoring the edge orientation the resulting subgraph t(a,r) is a

2-tree of G, and node a serves as reference node in one subgraph and node r in the other of t'(a;r). Similarly, let t'(a,b;r) and t'(a,b;r,c) and t'(a,b,c;r) be type-II, type-III and type-IV 2-trees of G', respectively. Then their nonoriented counterparts are 2-trees of type-II, type-III and type-IV of G, and node a serves as reference node in one subgraph and node r in the other.

E. FUNDAMENTAL SETS OF LOOPS

Consider any tree of a connected graph; since it is connected there is a tree path between any two nodes and, since it contains no circuits, this tree path is unique. Now consider the tree with one of its chords. Between the two nodes terminating the chord there is a unique tree path which, together with the chord, defines a circuit of the graph. Similarly each of the (b-n+1) chords of the tree defines a circuit of the graph; these (b-n+1) circuits are called a fundamental set of loops of the graph; their orientations are defined to coincide with those of their defining chords.

Definition 2.11 A fundamental loop matrix with respect to a tree of a connected graph G is the matrix B_f of order (b-n+1) x b with each row identified by a fundamental circuit (with respect to the tree) and each column by an edge such that

$$B_f = \left[b_{ij}\right] (b-n+1) \times b$$

where

$$b_{ij} = \begin{cases} 1, \text{ if edge } j \text{ is in circuit } c_i \text{ and the orientation of } j \text{ coincides with the orientation of } c_i \\ -1, \text{ if edge } j \text{ is in } c_i \text{ but the orientation of } j \text{ is opposite to the orientation of } c_i \\ 0, \text{ if } j \text{ is not in } c_i. \end{cases}$$

Let the edges of the graph are numbered so that the chords of the tree form the first b-n+1 edges and, if the fundamental loops are numbered correspondingly, B_f will be of the form:

$$B_f = [U_m \; B_{f12}]$$

where m = b-n+1. Let the columns of the reduced matrix A be arranged in the same order of edge:

$$A = [A_{11} \; A_{12}]$$

The following important interrelationships between matrices A_{11}, A_{12}, B_{f12} have been pointed out.

Property 2.11 $B_{f12} = -[A_{12}^{-1} A_{11}]$

This property results directly from Property 2.4.

Property 2.12 B_f is totally unimodular.

This property has been proved by Belevitch in [1].

The inverse of A_{12} for a connected graph has been given an interesting interpretation by Branin [2] as follows:

Property 2.13 For a connected graph the element a'_{ij} of the inverse of A_{12} is:

$$a'_{ij} = \begin{cases} +1, & \text{if branch } i \text{ of the tree is contained in the unique tree path from node } j \text{ to the reference node and is oriented from node } j \text{ to the reference node} \\ -1, & \text{if branch } i \text{ of the tree is contained in the unique tree path from node } j \text{ to the reference node and is oriented from the reference node to node } j \\ 0, & \text{if branch } i \text{ of the tree is not contained in the unique tree path from node } j \text{ to the reference node.} \end{cases}$$

F. CIRCUIT-SPACE AND INDEPENDENT CIRCUITS

Definition 2.12 Let the rank of B_a be $m = b-n+1$. The circuit-space of a graph is defined as the vector-space spanned by the rows of B_a over the field of real numbers. This space, denoted by $\underline{L}$, is obviously an m-dimensional subspace of R^b, and any set of m linearly independent rows of B_a will serve as basis for $\underline{L}$. Let any m x b matrix formed by m independent rows of B_a be denoted by B. Then the set of m circuits, which are individually associated with each row of B, is called a set of independent circuits.

From Definition 2.8, the following properties of the independent loop matrix can be proved.

Property 2.14 $AB^T = BA^T$.

This property is obvious from Property 2.4.

Property 2.15 There is a non-singular m x m matrix T of 1, -1 and 0 such that

$$B = TB_f$$

where B_f is a fundamental circuit matrix of the graph.

This property results from the observation that the rows of B and the rows of B_f span the same space, and since B_f contains the unit matrix.

Property 2.16 An m-minor of B is non-singular if and only if the m columns of the minor correspond to the branches of a cotree.

The proof of this property has been given by Bryant in 3 .

G. SEGS AND THE SEG MATRIX

Definition 2.13 Let N_1 be a non-empty proper subset of the set of nodes N of a graph, and let $N_2 = N - N_1$. The set of branches, each of which is incident with one node in N_1 and one node is N_2, is called a seg of the graph.

The following properties of seg can be deduced from Definition 2.13.

Property 2.17 The set of branches incident at a node is a seg.

Property 2.18 A connected graph contains $2^{n-1} - 1$ different non-empty segs.

Property 2.19 Any seg contains an even number of edges in common with any circuit, and half of which the orientations of seg and circuit will agree while on the other half their orientations will oppose.

The proofs of properties 2.18 and 2.19 have been given in [3], by Bryant.

Definition 2.14 The seg matrix, C_a, is the matrix of order s x b with each row identified by a seg and each column by an edge, such that

$$c_{ij} = \begin{cases} +1, & \text{if edge } j \text{ is incident with seg } j \text{ and has the same orientation} \\ -1, & \text{if edge } j \text{ is incident with seg } j \text{ and has the opposite orientation} \\ 0, & \text{if the edge } j \text{ is not incident with seg } j. \end{cases}$$

From Definition 2.14, the following properties of the seg matrix can be proved.

Property 2.20 $B_a C_a^T = 0$.

Property 2.20 follows directly from Property 2.19.

Property 2.21 The rank of the seg matrix is n-1.

Proof: From Property 2.20, rank of $C_a \leq n-1$; from Property 2.17: $C_a > A$, and rank $(A_a) = n-1$. Hence Property 2.21.

H. FUNDAMENTAL SETS OF SEGS

Definition 2.15 Consider a tree j of a tree t of a connected graph. Branch j segregates the nodes of G into two sets; the corresponding seg of the graph consists of that particular tree branch together with certain of the chords of the tree. The set of n-1 seg obtained in this way from a tree has the property t that each seg contains one branch which is not contained in any of the other seg of the set. This set of n-1 segs is called a fundamental set of segs. The orientation of these segs is defined to agree with the orientation of the defining tree branch.

Definition 2.16 A fundamental seg matrix, denoted by C_f, is a submatrix of order (n-1) x b and rank n-1 formed by n-1 fundamental segs of C_a.

Let the tree branches be numbered so that the last n-1 columns in C_f correspond to the tree branches and let the segs be numbered correspondingly, so that C_f can be written as:

$$C_f = [E \; U_{n-1}]$$

With B_f and A written in the corresponding forms

$$B_f = [U_m \; F]$$

$$A = [A_{11} \; A_{12}]$$

The following interrelationships have been proved in [3].

Property 2.22 $E = -F^T = A_{12}^{-1} A_{11}$ and $C_f = A_{12}^{-1} A$.

Property 2.23 Those chords which combine with a given tree branch to form the corresponding seg are precisely those chords whose fundamental circuits contain the tree branch.

Property 2.24 C_f is totally unimodular.

I. SEG-SPACE AND INDEPENDENT SEGS

Definition 2.17 Let the rank of C_a be n-1. The seg-space of a graph is defined as the vector-space spanned by the row of C_a over the field of real numbers.

This space, denoted by $\underline{C}$, is an (n-1)-dimensional subspace of R^b; any set of n-1 linearly independent rows of C_a will serve as a basis for $\underline{C}$. Any (n-1) x b submatrix formed

from n-1 independent rows of C_a, denoted by C, is called a set of independent segs of the graph.

Since C is a submatrix of C_a, the following properties of C are obvious.

Property 2.25 $BC^T = 0$ and $CB^T = 0$.

Property 2.26 There is a non-singular (n-1) x (n-1) matrix S whose element are 1, -1 and 0 such that

$$C = S\, C_f$$

Property 2.27 An (n-1) minor of C is non-singular if and only if the n-1 columns of the minor correspond to the edge of a tree.

J. CUT-SET

Definition 2.18 A cutset of a connected graph is a minimal set of edges whose deletion will separate the remainder of the graph into two disjoint subgraphs, one of which may be an isolated node. If one isolated node results, the cutset is called a vertex cutset.

From Definition 2.18, the following properties can be proved.

Property 2.28 A cutset is a seg.

Property 2.29 A seg is either a cutset or a disjoint union of cutsets.

Property 2.30 Each fundamental seg is a cutset.

K. ORTHOGONALITY OF SEG-SPACE AND CIRCUIT-SPACE

Consider the circuit-space $\underline{L}$ and the Seg-space $\underline{C}$ of a graph. The following results have been proved in [3].

Property 2.31 The circuit-space and the seg-space form orthogonal sub-space of R^b.

This property is obvious from Property 2.4.

Property 2.32 Any vector X from R^b may be expressed uniquely as the sum of two vector X_L and X_C with $X_L \varepsilon \underline{L}$ and $X_C \varepsilon \underline{C}$.

Property 2.33 Any vector X R^b which satisfies CX = 0 is contained in $\underline{L}$, and any vector Y R^b which satisfies BY = 0 is contained in $\underline{C}$.

L. PATHS AND THE PATH MATRIX

Definition 2.19 A path is a connected ordered sequence of edges whose terminal nodes (the first and the last relative to the ordering) are of degree 1 and whose terminal vertices are of degree 2.

Definition 2.20 A path matrix, P, of a connected graph is a matrix whose b columns correspond to the edges of the graph, and whose row correspond to all paths between the first vertex s and the last vertex t. The entry p_{ij} of P is defined as follows:

$$p_{ij} = \begin{cases} 1, & \text{if edge } j \text{ is in path } i \\ 0, & \text{if edge } j \text{ is not in path } i \end{cases}$$

Property 2.34 Let A be the reduced incidence matrix of which the columns are arranged in the same order as in the

path matrix P, then

$$AP^T = H \quad \text{(mode.2)}$$

where $H = h_{ij}$ with

$$h_{ij} = \begin{cases} 1, \text{ if } i = s \text{ or } t \text{ for any } j \\ 0, \text{ otherwise.} \end{cases}$$

and s and t are terminal nodes of the paths from which P is defined.

This property has been proved by Okada [22]. Other interesting properties of the path matrix have been given in [33].

III. THE EDGE T-MATRIX AND NETWORK REPRESENTATION

A. INTRODUCTION

The main purpose of this chapter is to introduce the edge T-matrix which is used as a new tool for network representation. Three different definitions of the edge T-matrix, which are only slightly different, are given to cover respectively all types of graphs, namely, the non-oriented graph, oriented or directed graphs. General properties of the edge T-matrix will be pointed out. Next, the C-transformation will be defined and used for generating the set of subsequent edge T-matrices. Several properties of the C-transformation and of the set of subsequent edge T-matrices will also be proved.

The edge T-matrix and the C-transformation defined in this chapter can be considered as a generalization of the 'T-triangle' which has been introduced by Chang [6] for deriving a fast tree finding algorithm, for non-oriented graphs.

B. DEFINITION OF THE EDGE T-MATRIX

Depending on the problem under consideration, a network can be associated with a non-oriented or an oriented graph, or a digraph. Therefore in the following three distinct definitions of the edge T-matrix associated with a non-oriented graph G_o, and oriented graph G_o', or a directed graph will be given. The use of these edges T-matrices

will be pointed out in the next chapters for specified problems.

Definition 3.1-a Let the nodes of a directed graph G_o, be numbered consecutively from 1 to n, and the edges from 1 to b. The edge T-matrix $|T_1$ associated with G_d is a triangular array of which the i^{th} row corresponds to node i+1 and the j^{th} row column to node j. The entry t_{ij} in the i^{th} row and the j^{th} column is:

$$t_{ij} = \begin{cases} 0, \text{ if nodes } i+1 \text{ and } j \text{ are disconnected,} \\ \text{a binomial } (i+1 y_{i+1}) + (j, y_j) \text{ if nodes } i+1 \text{ and } j \text{ are connected.} \end{cases}$$

where (k, y_k) is an ordered pair of which y_k denotes the sum of all edge designations of the edges connecting nodes i+1 and j, and having node k as initial node, with K = i+1,j.

Definition 3.1-b In the case of a non-oriented graph G_o, the entry t_{ij} is simply:

$$t_{ij} = \begin{cases} 0, \text{ if nodes } i+1 \text{ and } j \text{ are disconnected} \\ \text{sum of all edge designations of the edges between nodes } i+1 \text{ and } j. \end{cases}$$

Definition 3.1-c If the graph is oriented, the entry t_{ij} is defined exactly as in Definition 3.1-b, but the edge designations are:

"+" if the edge is oriented away from node j.

"-" if the edge is oriented towards node j.

Definitions 3.1 will be illustrated in the following example.

Example 3.1 Find the edge T-matrices associated with the graph shown in Figure 3.1.

(a) The edge T-matrix associated with the graph G_o is:

$$\underline{|T_1|} = \begin{array}{c|cccc} 2 & 1 & & & \\ 3 & 2 & & & \\ 4 & 0 & 4+7 & 5 & \\ 5 & 0 & 0 & 6 & 8 \\ \hline & 1 & 2 & 3 & 4 \end{array}$$

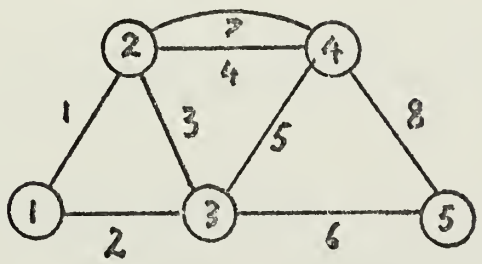


Figure 3.1.a. A Nonoriented Graph.

(b) The edge T-matrices associated with the oriented graph G_o, is:

By definition 3.1-a:

$$\underline{|T_1|} \quad \begin{array}{c|cccc} 2 & (1,1)+(2,2) & & & \\ 3 & (3,3) & (2,8) & & \\ 4 & (0,0) & (2,9)+(4,9) & (4,6) & \\ 5 & (1,4) & (0,0) & (3,10)+(5,11) & (4,7) \\ \hline & 1 & 2 & 3 & 4 \end{array}$$

By definition 3.1-c:

$$\underline{|T_1|} = \begin{array}{c|cccc} 2 & 1-2 & & & \\ 3 & 3 & -8 & & \\ 4 & 0 & 5-9 & 6 & \\ & 4 & 0 & 11-10 & -7 \\ \hline & 1 & 2 & 3 & 4 \end{array}$$

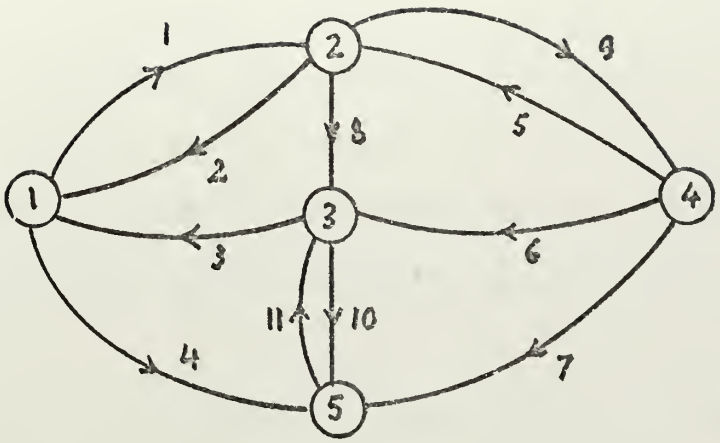


Figure 3.1-b. An Oriented Graph, $G_{o'}$.

C. PROPERTIES OF THE EDGE T-MATRIX

In light of Definitions 3.1, some general properties of the edge T-matrix will now be pointed out.

Property 3.1 Given the edge T-matrix, T_1, associated with a graph G. The set of edges of G which are incident at

(a) node 1 is represented by the set of non zero entries in column 1 of T_1,

(b) Node n is represented by the set of non zero elements in row n-1 of T_1,

(c) node i+1 is represented by the set of non zero elements in row i and column i+1 of T_1.

Proof: The subscript j of each entry t_{ij} in the first column is 1. This implies that the edges represented by this entry has one end connected to vertex 1. Similarly the subscript i of each non zero entry t_{ij} in the row (n-1) is n-1; then the edges of graph corresponding to this entry

has one end connected to node n. More general, each entry t_{ij} in the i^{th} row has a subscript i, then the edges of G associated with this entry has one end connected with node i+1, and each non zero entry in the $(i+1)^{th}$ column, has the subscript j = i+1, then edges of G associated with this entry has one end connected with node i+1. Hence Property 3.1.

Remark 3.1 This property is common for all forms of the edge T-matrices associated with a graph using either one of the Definitions 3.1.

Property 3.2 Let $\lfloor T_1$ be an edge T-matrix associated with a non directed graph G. Consider a set of n-1 entries, taken one from each row of $\lfloor T_1$ at a time, suppose that each of these entries corresponds to only one edge of G, then the product of these entries is the tree product of a tree of G.

Proof: An entry in row i represent an edge which connects node i+1 to one of the nodes preceding i+1. Then the set of n-1 entries, taken one from each row at a time, represents a set of n-1 edges of G which connect n nodes of G. Thus the product of these n-1 entries is clearly a tree product.

Definition 3.2 Given an edge T-matrix $\lfloor T_m$ associated with a non directed graph G. The partial tree summation obtained from $\lfloor T_m$ is:

$$|T_m|_p = \prod_{i=1}^{n-1} s_i^m \qquad (3.1)$$

where s_i^m is the sum of all entries in row i of $\underline{T_m}$,

Definition 3.2 can be justified by noting that each term in the right hand side of Equation 3.1 is a tree product of a tree of G, since it is formed by n-1 factors, of which each factor is a tree branch by applying Property 3.2.

In order to define the partial tree summation obtained from the edge T-matrix associated with a digraph, the following algebraic formulation is proposed.

Definition 3.3 The product of two ordered pairs (i,y_i) and (j,y_j) is an ordered pair (p,y_p) defined as follows:

$$(p,y_p) = (i,y_p)*(j,y_p)$$

where

$$p = \begin{cases} 0, & \text{if } i\ j = 0 \\ i\ j & \text{if } i\ j \neq 0 \end{cases}$$

$$q = \begin{cases} y_i y_j & \text{if } i\ j = 0 \\ 0, & \text{if } i\ j \neq 0 \end{cases}$$

From Definition 3.3, it can be easy to verify that the *product is commutative, associative and distributive with respect to addition.

Definition 3.4 Given an edge T-matrix $\underline{|T_m}$ associated with a directed graph G_d. Let each row, i, of $\underline{|T_m}$ be associated with a polynomial

$$S_i^m = \sum_{j=1}^{i} (j,y_j) \qquad (3.2)$$

The partial directed tree summation obtained from $\underline{|T_m}$ is defined by

$$|T_m|_{p^*} = \left| \prod_{i=1}^{n-1 *} S_i^m \right| \qquad (3.3)$$

where the right hand side of Equation represents the second element of the ordered pair

$$(p, y_p) = \prod_{i=1}^{n-1 *} S_i^m \qquad (3.4)$$

obtained by expanding the *multiplication.

Definition 3.4 can be justified by observing that:
(a) each term in the right hand side of Equation 3.3 is the tree product of a tree of G_d, and
(b) each tree obtained is a directed tree since the *multiplication discards all trees of G_d which do not satisfy Definition 2. , namely all trees in which there are more than one outgoing edges from a node.

Property 3.3 There is no duplication in the set of trees listed in the tree summations defined in Definitions 3.3 and 3.4.

Proof: There is a duplication if two obtained trees are formed by the same set of n-1 edges of the graph. From Definitions 3.3 and 3.4, there are no identical terms in Equations 3.1 and 3.3 unless the same edge is associated with two distinct entries of $\lfloor T_m$ which are located in two different rows in $\lfloor T_m$. This is impossible, since from Definition 3.1 there is an one-to-one correspondence between non zero entries of $\lfloor T_m$ and edges of the graph. Hency Property 3.3.

Property 3.4 If all the entries of row i, for $i \geq 1$, and those directly below row i, down to row n-1 of an edge T-matrix $\lfloor T_m$ are zero, then the graph associated with $\lfloor T_m$ is disconnected.

Proof: By Property 3.1, the edges which can be connected to one of the nodes preceeding node i+1 are those edges represented by non zero entries in row i^{th} or directly below them. Since all of these entries are zero, then the graph corresponding to $\lfloor T_m$ is disconnected.

Property 3.5 Given a value of K, with $1 \leq K \leq n-1$. A seg S_K, determined by partitioning the set of nodes of the graph into two subsets, say N_1 with K nodes, and N_2 with the remaining n-k nodes, can be obtained by taking all of the non zero entries t_{ij}, with $K \leq i \leq n-1$ and $1 \leq j \leq K$, in the edge T-matrix associated with the graph.

Proof: Replacing all non zero entries t_{ij} for $K \leq i \leq n-1$ and $1 \leq j \leq K$ by zero. The resulting edge T-matrix will be associated with a disconnected graph by virtue of Property 3.4. Then the set S_K defined above is clearly a seg of the graph by Definition 2.9. In particular, for K = 1, the seg is a node cutset, also is the seg S_{n-2}. In addition, if the subgraphs associated with the sub-edge T-matrices formed by the first K-1 rows and the last n-K-1 column respectively, are not disconnected then S_K is a minimal cutset.

Property 3.6 Consider the product

$$P = t_{kl} \prod_{i=1}^{k} t_{ii} \tag{3.1}$$

where t_{ij} are entries of an edge T-matrix $|\underline{T}$.

a) If $|\underline{T}$ is associated with a non oriented graph and if none of the entries in the right hand side of Eq. 2.1 is zero, then P is the path product of the path whose terminal nodes are 1 and k, and its interior nodes are $2 \leq i \leq k-1$.

b) If $|\underline{T}$ is associated with an oriented graph and if none of the entries in the right hand side of Eq. 3.1 is zero and all of the entries t_{ii} have the same sign which is opposite to that of t_{kl}, then P is the path product of the oriented path connecting node 1 to node k with nodes i, with $2 \leq i \leq k-1$ as interior nodes, whose orientation is defined by that of t_{kl}.

Proof: From Definiti9n 3.1, it is clear that each entry under the multiplication sign represents an edge with connects node i+1 to node i+2, and t_{kl} connects node 1 to node k. Hence Property 3.6-a. If in addition, suppose that all entries t_{ii} have the same sign, say +, and t_{kl} is of sign -, then the path is clearly oriented from node 1 to node k by the light of the sign convention given in Definition 3.1-c.

Property 3.7 Consider the product

$$C = t_{k-1,1} \prod_{i=1}^{k-1} t_{ii} \tag{3.2}$$

where t_{ij} are entries of an edge T-matrix $|\underline{T}$

a) If $|\underline{T}$ is associated with a non oriented graph and if none of the entries in the right hand side of Eq. 3.2 is zero, then C is the circuit product of a circuit formed by k edges with interior nodes i, where $1 \leq i \leq k-1$.

b) If $|\underline{T}$ is associated with an oriented graph, and if none of the entries in the right hand side of Eq. 3.2 is zero and if all of the entries t_{ii} have the same sign which is opposite to that of $t_{k-1,1}$, then C is the circuit product of the oriented circuit whose interior nodes are nodes i, with $1 \leq i \leq k-1$, and whose orientation is defined by that of $t_{k-1,1}$.

Proof: Property 3.7 is a corrollary of Property 3.6. The expression of C can be rewritten as

$$C = t_{11} \; t_{k-1,1} \prod_{i=1}^{k-1} t_{ii} \qquad (3.2\text{-a})$$

By Property 3.6, the expression in the bracket is the path product of a path of length k-1, connecting node 1 to node k-1, and edge t_{11} corresponds to the edge that connects the two terminal nodes of the path. Hence Property 3.7 is proved.

D. DEFINITION OF THE C-TRANSFORMATION

Definition 3.5-a Given the edge T-matrix $|\underline{T_1} = \{t_{pq}\}$ associated with a graph G by applying Definitions 3.1-a or 3.1-b. It is said that the edge T-matrix $|\underline{T_i} = \{t'_{pq}\}$ is derived from $|\underline{T_1}$ by applying the C-transformation using column $j \geq 2$ as operating column, if:

(1) $t'_{pq} = t_{pq}$ for $1 \leq p,q \leq j-2$

(2) $t'_{pq} = t_{p+1,q}$ for $j-1 \leq p \leq n-2$ and $1 \leq q \leq j-1$

(3) $t'_{pq} = t_{p+1,q+1}$ for $j \leq p \leq n-2$ and $j \leq q \leq n-2$

(4) $$t'_{n-1,q} = \begin{cases} t_{j-1,q} & \text{for } 1 \leq q \leq j-1 \\ t_{qj} & \text{for } j \leq q \leq n-1 \end{cases}$$

In particular for j=1, then:

(5) $t'_{pq} = t_{p+1,q+1}$ for $1 \leq p,q \leq n-2$

(6) $t'_{n-1,q} = t_{q1}$ for $1 \leq q \leq n-1$

Remark 3.2 Let G_j be the graph associated with $\lfloor T_{1j}$. Then from Definition 3.1, it follows that G_j can be obtained from G by rearranging the nodes of G in the following sequence:

$$1,2,\ldots,j-1,j+1,j+2,\ldots,n-1,n,j.$$

Definition 3.5-b Given the edge T-matrix $\lfloor T_1 = \{t_{pq}\}$ associated with an oriented graph G_o, by applying Definition 3.1-c. The edge T-matrix $\lfloor T_{1j} = \{t'_{pq}\}$ is said derived from $\lfloor T_1$ by applying the C-transformation with respect to column $j \geq 2$, if the entries t'_{pq} are obtained from t_{pq} by using Equations (1) through (4) and in Equation (4) replace t_{qj} by its negative, $-t_{qj}$.

For j = 1, using Equation (5) and 6, but in Equation 6, replace t_{q1} by its negative $-t_{q1}$.

Definition 3.6-a and 3.6-b will be illustrated by the following example.

Example 3.2 Find the edge T-matrix $|\underline{T_{13}}$ derived from the edge T-matrix given in Example 3.1.

(a) Suppose that $|\underline{T_1}$ has been obtained by applying Definition 3.1-a. The edge T-matrix $|\underline{T_{13}}$ derived from $|\underline{T_1}$ using column 3 as operating column is:

$$|\underline{T_{13}} = \begin{array}{c|cccc} 2 & (1,1)+(2,2) & & & \\ 4 & (0,0) & (2.9)+(4.9) & & \\ 5 & (1,4) & (0,0) & (4.7) & \\ 3 & (3,3) & (2,8) & (4,6) & (3,10)+(5,11) \\ \hline & 1 & 2 & 4 & 5 \end{array}$$

(b) Suppose that $|\underline{T_1}$ has been obtained by applying Definition 3.1-c. The edge T-matrix $|\underline{T_{13}}$ derived from $|\underline{T_1}$ using column 3 as operating column is:

$$|\underline{T_{13}} \quad \begin{array}{c|cccc} 2 & 1-2 & & & \\ 4 & 0 & 5-9 & & \\ 5 & 4 & 0 & -7 & \\ 3 & -3 & 8 & -6 & -11+10 \\ \hline & 1 & 2 & 4 & 5 \end{array}$$

1. Properties of the C-transformation

Property 3.8 Let $|\underline{T_{1j}}$ be the transform of a given edge T-matrix $|\underline{T_1}$ obtained by applying the C-transformation with respect to column j of $|\underline{T_1}$. Let G_{1j} and G_1 be the graphs associated with $|\underline{T_{1j}}$ and $|\underline{T_1}$ respectively. Then G_{1j} and G_1 are 2-isomorphic.

Proof: From Remark 3.2, the process for generating $|\underline{T_1}$ from $|\underline{T_1}$ results from a rearrangement of the sequence of

vertices of G_1, then clearly these two graphs are 2-isomorphic.

Property 3.9 If the C-transformation is performed on a given edge T-matrix $\lfloor T_1$ using column j as operating column for n-j+1 times consecutively, then the final newly generated edge T-matrix will be identical to the original edge T-matrix $\lfloor T_1$.

Proof: From remark 3.2, the sequence of node of the graph associated with T_{1j} can be obtained from that of $\lfloor T_1$ by a circular substitution performed on the subsequence of nodes

$$(j, j+1, \ldots, n-1, n)$$

while the first j-1 elements remain unchanged. Since there are n-j elements in this subsequence of nodes, then if the substitution operation is repeated n-j+1 time consecutively, then the final subsequence will be identical with the original. Hence Property 3.12.

2. Complete Set of Subsequent Edge T-matrices

Given a graph of n vertices, a set of n! 2-isomorphic graphs can be obtained by considering n! distinct arrangements of the sequence of vertices. It is said that

Definition 3.6 The set of edge T-matrices associated with the set of n! 2-isomorphic graphs derived from a given graph by considering n! distinct arrangements of the sequence of nodes of the original graph is called the complete set of

subsequent edge T-matrices derived from the edge T-matrix associated with the original graph.

Property 3.10 The complete set of subsequent edge T-matrices derived from a given edge T-matrix can be obtained from the original edge T-matrix as follows:

1) Applying the C-transformation with respect to column j, with $1 \leq j \leq n-1$,

2) From each newly generated edge T-matrix obtain a set of edge T-matrices by applying the C-transformation with respect to column k, with $j \leq k \leq n-1$,

3) Repeat (2) on each newly generated edge T-matrices to obtain successively all members of the set.

Proof: From Property 3.12, each newly generated edge T-matrices is associated with a 2-isomorphic graph of the given graph, and by the process outlined in (2) and (3) a column j is used as operating column exactly n-j times consecutively, then no two edge T-matrices generated by the process are associated with two identical sequences of nodes by virtue of Property 3.12. In addition the process given from (1) to (3) includes circular substitutions of order from 1 to n; then it has been proved that [17] these circular substitutions represent the whole of the permutations of the n nodes in the sequence. Hence Property 3.10 is proved.

3. Subset of Subsequent Edge T-matrices

Definition 3.6 Let G be a given graph, whose sequence of node is denoted by S. Let S(n-p-1) be the set of (n-p-1)! distinct sequences of node derived from S by performing the

circular substitution operation of order p on S. Consider the set of 2-isomorphic graphs derived from G whose sequence of nodes are one of the members of S(n-p-1). The set of edge T-matrices associated with these graphs is called the p-subset of subsequent edge T-matrices derived from the original edge T-matrix associated with G.

Repeating the proof of Property 3.10, it follows that:

Property 3.11 The p-subset of subsequent edge T-matrices derived from a given edge T-matrix can be obtained by

1) Applying the C-transformation with respect to column j, with $p+1 \leq j \leq n-1$

2) From each newly generated edge T-matrix, obtain a set of edge T-matrices by applying the C-transformation with respect to column k, with $j \leq k \leq n-1$

3) Repeat (2) on each newly generated edge T-matrix to obtain successively all members of the subset.

IV. THE EDGE T-MATRIX AND OTHER MATRICIAL REPRESENTATIONS OF NETWORK GRAPHS

A. INTRODUCTION

In this chapter, techniques for deriving the following matricial representation of network graphs from its edge T-matrix will be presented:

1. A fundamental loop or cutset matrix,
2. A path matrix,
3. The circuit matrix,
4. The seg matrix

Problem 1 has been studied by Welch [32] and Gotleib and Corneil [14]. However these solutions suffer from a major disadvantage: Welch's procedure needs storage for nxm incidence matrices and two vectors of length m; Gotleib and Corneil's solution requires storage for nxn adjacency matrices, where n is the number of nodes of the graph and m that of its edges.

Problems 2, 3, and 4 are usually attacked in parallel. These problems have been studied by Parthasarathy [24], Cartwright and Gleason [4], Kamea [15], Maxwell and Reed [19] and recently by Char [7]. These existing solutions can be classified, according to their approaches, into two classes. Parthasarathy, Cartwright and Gleason, and Kamea have used the connection matrix. Maxwell and Reed's method has been based on the ring sum operation performed on rows of a fundamental circuit matrix of the graph. Char's

solution is an improvement of Maxwell and Reed's algorithm and based upon an edge-numbering convention.

However each of the existing methods for solving the above mentioned problems suffers from one or more of the following disadvantages: (a) the method is too complicated for hand calculation or for computer implementation; (b) the solution involves duplications; (c) large computer memory is required; and the process requires long computation time.

Algorithm for obtaining a fundamental loop matrix will be presented in Section 4.2. It will be proved that the temporary storage needed is smaller than that required for two edge T-matrices. The computation time is minimized due to the fact that the process lists, without test, every loop directly from a resolving tree.

Algorithms for determining a path matrix, circuit matrix and seg matrix will be proposed in Sections 4.3, 4.5 and 4.6. It will be proved that no duplication occurs, then computation time can be minimized since no test is required, and the temporary storage need in the process is in maximum n - 1 edges T-matrices.

B. ALGORITHM FOR DETERMINING A FUNDAMENTAL LOOP MATRIX

To obtain a set of fundamental loops of a network is relatively an easy problem. This can be performed by choosing a tree, then with respect to every chord of this tree list the corresponding fundamental loop by considering a path, formed by tree branches, connecting terminals of that

chords. In order to use the edge T-matrix for solving this problem, the concept of resolving edge T-matrix will be first introduced.

1. The Revolving Edge T-Matrix

Definition 4.1 A column j of an edge T-matrix T is called a R-operating column if:

1) $t_{j-1,j-1} = 0$, and

2) $\sum_{i=j}^{n-1} t_{i,j-1} \neq 0$.

Definition 4.2 An edge T-matrix is called a resolving edge T-matrix if no column of this edge T-matrix can be used as an R operating column.

Property 4.1 Any edge T-matrix can be transformed into a resolving edge T-matrix.

Proof: Consider an arbitrary edge T-matrix $\lfloor T$. If all of its entries t_{ii} are non zero, then it is a resolving edge T-matrix. If some of the entries t_{ii} is equal to zero, then two cases are to be considered: (1) Suppose that $t_{ii} = 0$, and all of the entries in column i are also zero, then column i+1 can not be used as an R-operating column. If this condition is satisfied for all $t_{ii} = 0$, then T is a resolving edge T-matrix. (2) Suppose now that $t_{ii} = 0$, but $t_{pi} \neq 0$, then column i+1 can be used as an R-operating column. Applying the C-transformation with respect to column i+1 consecutively p-i times, the entry t_{ii} in the resulting edge T-matrix will become non zero, then column i+1 will not be longer an

R-operating column. Applying this process, starting from the most left column of which the entry t_{ii} is zero, then repeat the process for the resulting edge T-matrix of which the column i+1 is not an R-operating column, until no column of the newly transformed edge T-matrix can be used as an R-operating column, then the final edge T-matrix will be transformed into a resolving edge T-matrix.

Definition 4.3 Given a resolving edge T-matrix $\underline{|T_r}$ of a graph G, of no parallel edge. The resolving tree of G defined from $\underline{|T_r}$ is the tree whose branches are associated with:

1) all non zero entries t_{ii}, and
2) the most right entry t_{iq} in each row i, if $t_{ii} = 0$.

From Definition 4.3, it is seen that the resolving tree of a graph G, defined from its resolving edge T-matrix $\underline{|T_r}$ is composed of some trains of tree branches which correspond individually with each subset of consecutive non zero entries t_{ii} of $\underline{|T_r}$. These train of tree branches are connected together by attachment tree branches which are associated with the tree branches associated with the selected entries t_{iq}. With respect to a resolving tree, the set of chord defined relatively to this tree can be classified into three classes as follows.

Definition 4.4 With respect to a resolving tree, a chord is said to be a

(1) Type-I chord, if both of its terminal nodes are in the same train of tree branches.

(2) Type-II chord, if it has only one terminal node in a train of tree branches and the other is connected to an attachment tree branch, which in turn is connected to the same train of tree branches.

(3) Type-III chord, if it has each terminal node in a different train of tree branches.

2. Fundamental Loop Listing Algorithm

The concept of classification of chords defined with respect to a resolving tree, together with technique for obtaining the circuit products corresponding to every loop of a fundamental set of circuits will be better illustrated by use of an example.

Example 4.1 Find the set of fundamental loop from the resolving edge T-matrix given below.

$$\lfloor T_r = \begin{array}{c|cccccccccccccc} 2 & a & & & & & & & & & & & & & \\ 3 & . & b & & & & & & & & & & & & \\ 4 & . & . & c & & & & & & & & & & & \\ 5 & p & . & . & d & & & & & & & & & & \\ 6 & q & . & e & . & . & & & & & & & & & \\ 7 & . & . & . & s & . & . & & & & & & & & \\ 8 & . & . & . & . & . & . & f & & & & & & & \\ 9 & . & . & . & w & . & . & . & g & & & & & & \\ 10 & . & . & z & . & . & . & r & . & h & & & & & \\ 11 & . & i & . & . & . & . & . & . & . & & & & & \\ 12 & u & . & . & . & . & . & . & . & . & j & & & & \\ 13 & . & . & . & . & . & . & . & . & . & . & k & & & \\ 14 & . & . & . & x & . & . & . & . & . & v & l & & & \\ 15 & . & . & . & . & . & . & . & m & . & . & . & . & & \\ 16 & . & . & . & . & . & . & . & . & . & . & . & . & n & \\ 17 & . & y & . & . & . & . & . & t & . & . & . & . & . & o \\ \hline & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \end{array}$$

The set of branches of the resolving tree is:

(a,b,c,d; e; s,f,g,h; i,j,k,l; m,n,o)

The resolving tree is formed by five trains of tree branches:

P_1 = (a,b,c,d); P_2 = (e); P_3 = (s,f,g,h); P_4 = (i,j,k,l) and P_5 = (m,n,o).

The set of Type-I chords defined with respect to the above trains of tree branches is composed of:

p, defined with respect to P_1

w,r defined with respect to P_2

v, defined with respect to P_3.

The set of type-II chords is composed of a single element q, defined with respect to P_1 and the attachment edge e.

The set of Type-III chords is:

z connecting P_1 and P_2,

u and x, connecting P_1 and P_3,

y, connecting P_1 and P_4,

t, connecting P_2 and P_4.

To find the circuit product of the circuit defined with respect to edge p, it is noted that p is of terminals nodes 1 and 5, in addition a path connecting these nodes can be obtained by the set of entries a, b, c, and d, taken one in each row, with the first element, a in the column containing p, and the last element d in the row that contains also p. This circuit product is:

$$P(p) = abcd\underline{p}$$

Similarly, for chords r and v:

$P(r) = fgh\underline{r}$, and $P(v) = kl\underline{v}$.

To find the circuit product of the circuit defined with respect to edge w, it is noted that w connecting nodes 4 and 9, which are nodes included in the train of tree branches P_2. The path connecting 4 and 9 is formed by s, f, and g, taken also on each row with the first element s in the column containing w and the last entry g in the row containing w also. Then

$P(w) = sfg\underline{w}$.

In general a Type-I chord circuit product can be expressed as:

$$P(t_{ij}) = t_{ij} \prod_{k=1}^{j} t_{kk} \qquad (4.1.a)$$

and

$$C(t_{ij}) = t_{ij} t_{xj} \left[\prod_{k=x+1}^{j} t_{kk} \right] \qquad (4.1.b)$$

if t_{xj} is a tree branch. Eq. 4.1.a is clearly a particular case of Eq. 4.1.b.

To find the circuit product associated with the loop defined with respect to type-II chord q, it is observed that q connecting nodes 1 and 6. The path which connects these two nodes can be obtained by the set of entries a,b. The first entry is taken in the column that contain q and the last in the column which preceeds the column that contains the attachment edge e. This circuit product is:

$C(q) = abe\underline{q}$.

In general, given a type-II chord t_{ij}, defined with respect to a train of tree branch P_k and let the attachment edge be t_{iq}, then the circuit product of the loop defined by t_{ij} is

$$C(t_{ij}) = t_{ij} t_{xj} \prod_{k=x+1}^{q-1} t_{kk}\, t_{iq} \qquad (4.2)$$

where t_{xj} is the tree branch of P_k which is located in the same column with t_{ij}, with $i > x$.

The determination of the circuit product of the loop defined with respect to chord of type-III, z, connecting P_1 and P_2, can be performed as follows. Consider a subgraph of the given graph which is formed only by nodes contained in P_1 and P_2 and edges between these nodes. For simplicity, type-I and type-II chords defined with respect to these trains of tree branches are also deleted. The edge T-matrix associated with this graph can be derived from $\lfloor \underline{T_r}$ by:

(1) letting entries of $\lfloor \underline{T_r}$ corresponding to type-I and Type-II chords, defined with respect to P_1 and P^2, equal zero,

(2) considering only rows and columns associated with nodes contained in P_1 and P_2.

(3) removing all rows and column associated with nodes of degree one, namely row r and column r+1 in which there is exactly one non zero element.

It is found that:

$$\underline{|T(P_1,P_2)} =$$

2	a							
3	.	b						
4	.	.	c					
5	.	.	.	d				
7	.	.	.	s	.			
8	.	.	.	.	.	f		
9	.	.	.	.	.	.	g	
10	.	.	z	.	.	.	.	h
	1	2	3	4	5	7	8	9

after (1) and (2) have been performed. Removing row 4 and column 5 which are associated with node 4, gives:

$$\underline{|T(P_1,P_2)} =$$

2	a						
3	.	b					
4	.	.	c				
7	.	.	.	s			
8	.	.	.	.	f		
9	.	.	.	.	.	g	
10	.	.	z	.	.	.	h
	1	2	3	4	7	8	9

The circuit product of the loop defined with respect to chords z can now be obtained by applying Eq. 4.1.a to this reduced edge T-matrix.

$$C(z) = csfghz$$

To determine the circuit product of the loop defined with respect to chord to type-II, u and x connecting P_1 and P_4, the operations (1), (2) and (3) carried out give:

$$\underline{|T(P_1,P_4)} =$$

2	a						
3	.	b					
4	.	.	c				
11	.	i	.	.			
12	u	.	.	.	j		
13	.	.	.	.	.	k	
14	.	.	.	x	.	.	l
	1	2	3	4	11	12	13

$\underline{|T(P_1,P_4)}$ is not a resolving matrix, in addition it is observed that there are two type-III chords, namely u and v, which are located in both sides of the column containing the tree branch i, which connects P_1 to P_4. It is clear that fundamental loops defined relative to u and x respectively, must contain the edge i. To evaluate the circuit products of fundamental loops formed by u and x, it is better to investigate the graph associated with the reduced matrix $\underline{|T(P_1,P_4)}$ shown in Figure 4.1. From this figure:

$$C(u) = a(i)j\underline{u}$$

and

$$C(x) = cb(i)kl\underline{x}.$$

This result can be generalized as follows. Let t_{gh} be the entry of the reduced edge T-matrix $\underline{|T(P_a,P_b)}$, corresponding to the edge connecting P_a and P_b. Let t_{ij} be the entry associated with a type-III chord defined with respect to P_a and P_b. Then

(1) for $j < h$ and $g < i$

$$C(t_{ij}) = t_{ij}\left[\prod_{k=j}^{h-1} t_{kk}\right] t_{gh}\left[\prod_{k=g+1}^{i} t_{kk}\right] \qquad (4.3.a)$$

and for $j > h$ and $g > i$

$$C(t_{ij}) = t_{ij}\left[\prod_{k=0}^{j-h} t_{j-k,j-k}\right] t_{gh}\left[\prod_{k=g+1}^{i} t_{kk}\right] \qquad (4.3.b)$$

To determine the circuit product of the fundamental loop defined with respect to edge t, it is noted that this edge

connecting P_3 and P_5. The reduced edge T-matrix associated with the subgraph formed by P_3 and P_5 is

$$\underline{|T(P_3,P_5)} = \begin{array}{c|cccccc} 7 & s & & & & & \\ 8 & . & f & & & & \\ 9 & . & . & g & & & \\ 15 & . & . & m & . & & \\ 16 & . & . & . & . & n & \\ 17 & . & . & . & t & . & o \\ \hline & 4 & 7 & 8 & 9 & 15 & 16 \end{array}$$

Applying Eq. 4.3.b leads to:

$$C(t) = gmnot$$

To determine the circuit product of the fundamental loop obtained with respect to edge y, it is noted that y connects P_1 to P_5, but there is no common node in these trains of tree branches. Furthermore, train of tree branch P_3 has common nodes with both P_1 and P_5. Then the reduced edge T-matrix from which the circuit product of the loop defined relatively with respect to y can be obtained is formed by three trains of tree branches, namely P_1, P_3 and P_5.

$$\underline{|T(P_1,P_3,P_5)} = \begin{array}{c|ccccccc} 3 & b & & & & & & \\ 4 & . & c & & & & & \\ 7 & . & . & s & & & & \\ 8 & . & . & . & f & & & \\ 15 & . & . & . & . & m & & \\ 16 & . & . & . & . & . & n & \\ 17 & y & . & . & . & . & . & o \\ \hline & 2 & 3 & 4 & 7 & 8 & 15 & 16 \end{array}$$

The circuit product of the fundamental loop defined with respect to edge y can now be obtained by applying Eq. 4.1.a to the reduced edge T-matrix $\underline{|T(P_1,P_3,P_5)}$.

$C(y) = bcsfmnoy$

The final fundamental loop matrix is:

$B_f =$

	p	r	v	w	q	z	u	x	y	t	a	b	c	d	e	s	f	g	h	i	j	k	l	m	n	o
	1	.	.	.	.	.	.	.	.	.	1	1	1	1	.	.	.	.	.	.	.	.	.	.	.	.
	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	1	1	.	.	.	.	.	.	.
	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	1	.	.	.
	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	1	1	1	.	.	.	.	.	.	.	.
	.	.	.	.	1	.	.	.	.	.	1	1	.	.	1	.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	1	.	.	.	.	.	1	.	.	1	1	1	1	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	1	.	.	.	1	.	.	.	.	.	.	.	1	1	.	.	.	.	.	.
	.	.	.	.	.	.	.	1	.	.	.	1	1	.	.	.	.	.	1	.	1	1	.	.	.	.
	.	.	.	.	.	.	.	.	1	.	.	1	1	.	.	1	1	.	.	.	.	.	.	1	1	1
	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	1	.	.	.	.	.	1	1	1

The obtained result does agree with that given by a direct investigation on the graph shown in Figure 4.1.a.

Remark 4.1 In computation the circuit products of chord of type-III connecting P_i and P_j, with $i < j$, it is observed that in general there is no tree branch connecting P_i and P_j. This can be seen by the fact that P_i and P_j do not have a node in common. In this case, as shown in the computation of the loop defined with respect to edge t, in Exp. 4.1, a train of tree branches P_{j-k} which has common node with both P_i and P_j is required for the derivation of the reduced edge T-matrix $\underline{|T(P_i,P_{j-k},P_j)}$. In general such a reduced matrix can be always obtained, since the resolving tree is by definition a connected graph, then a path formed by tree branches between a node of P_j and a node of P_i is always existed. Such a path will include edges of P_j, $P_{j-1},\ldots,P_{j-k},\ldots,P_i$ since in an edge T-matrix a node associated with a row i is connected to any node associated

with any row preceding row i. Therefore, the search for the nodes of the reduced edge T-matrix $\underline{|T(P_1,P_j)}$ can be thought of as the search of the common node between $P_j, P_{j-1}, P_{j-k}..., P_i$. Furthermore, it is noted that the reduced edge T-matrix $\underline{|T(P_i,P_j)}$ is always either a resolving edge T-matrix, if all members of S(i,j), the set of type-III chords defined with respect to P_i and P_j are located to the left of the column that contains the tree branch connecting P_i to P_{j-k}, or an edge T-matrix of the form of $T(P_3,P_5)$ in Example 4.1. This can be seen by the fact that in the reduced edge T-matrix $T(P_1,P_j)$ only type-III chords defined with respect to P_i, P_j are considered together with the tree branches of P_i, P_{j-k} and P_j, and all the tree branches are represented by entries t_{hh}, except for the entry representing the tree branch which connects P_i to P_{j-k} in the case where there are entries corresponding to some member of S(I,J) located to the right of the column containing the tree branch connecting P_i and P_{j-k}. Therefore, all circuit products of fundamental loops associated with type-III chords, members of S(i,j) can be obtained by applying, to the reduced edge T-matrix Eq. 4.1.a if it is of the form of a resolving edge T-matrix, or Eqs. 4.3.a and 4.3.b accordingly to the location of the entries representing members of S(i,j) relatively to column containing the tree branch connecting P_i to P_{j-k}, if the reduced edge T-matrix is not of the form of a resolving edge T-matrix.

The method used in Example 4.1 can now be generalized and summarized in the following algorithm.

3. Algorithm for Listing a Set of Fundamental Loops

A set of fundamental loops of a graph G can be obtained through application of the following steps.

Step 1. Associated with G its edge T-matrix $\lfloor T$.

Transform $\lfloor T$ into a resolving edge T-matrix $\lfloor T_r$.

Classify the set of chords defined with respect to the resolving tree, by application of Definition 4.4.

Determine the connection between two trains of tree branches, P_i and P_j, using Remark 4.1, for every S(i,j).

Step 2. Compute the circuit products of all of the fundamental loops defined by type-I and type-II chords, by use of Eq. 4.1 and Eq. 4.2. Delete all type-I and type-II chords from $\lfloor T_r$.

Step 3. Compute the circuit products of all of the fundamental loops defined by type-III chords, members of S(i,j), by:

(a) forming the reduced edge T-matrix $\lfloor T(P_i, P_j)$ and

(b) applying Eq. 4.1 or Eqs. 4.3a and 4.3.b accordingly to the form of the reduced edge T-matrix.

Step 4. Repeat Step 3 for every set S(i,j) obtained from Step 1.

It is obvious that the rate of convergence of the above algorithm depends on the number of trains of tree branches of the resolving tree. In the case where all of the entries t_{ii} of the resolving edge T-matrix are non zero, then all

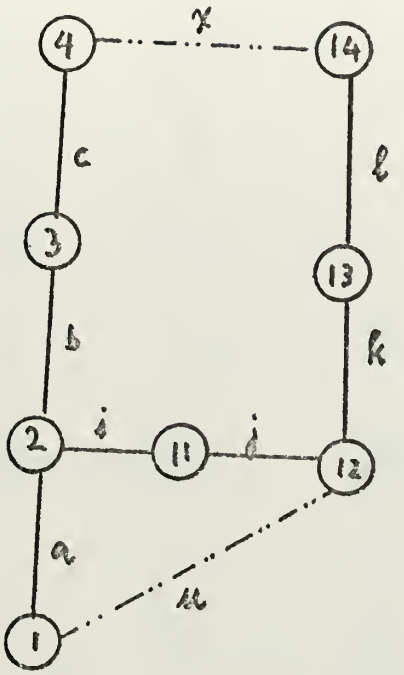


Figure 4.1. Illustration of the Process of Evaluating the Circuit Products of Fundamental Loops Associated with Type-III Chords.

chords of the resolving tree are type-I chords. Then no reduced edge T-matrix is required.

The computer storage needed for the implementation of the above algorithm is, in the worst case, that of two edge T-matrices. In addition, the computation time is minimized by the fact that all loops listed are fundamental loops, and no duplication occurs, then no check operation is necessary.

C. PATH LISTING ALGORITHM

In order to derive an algorithm for listing all possible paths between a pair of nodes in a given graph using edge

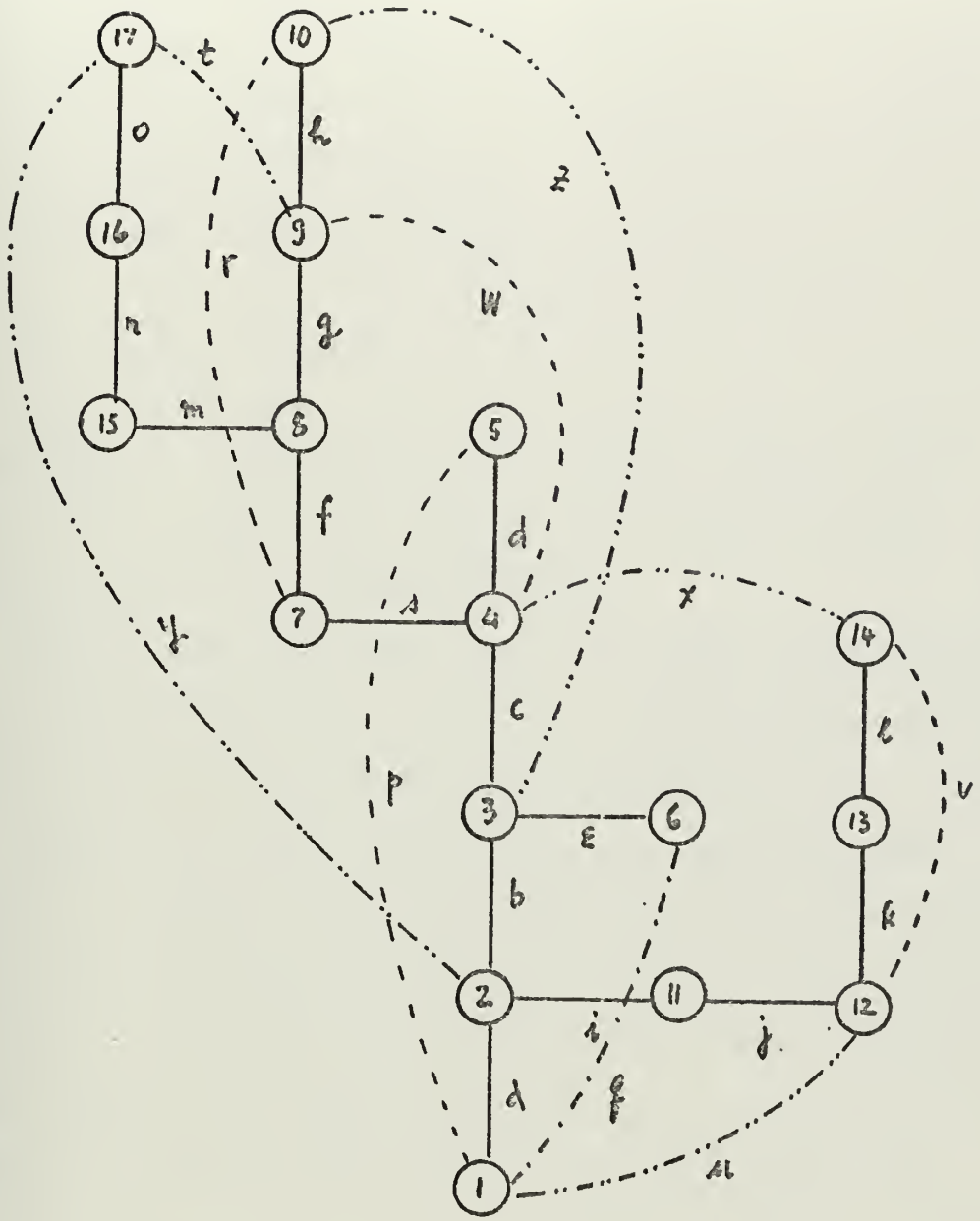


Figure 4.1.a. Illustration of the Process of Listing a Set of Fundamental Loops.

T-matrix and the C-transformation, we first define the P-operating column.

Definition 4.5 Any column $j \geq 3$ of an edge T-matrix is called a P-operating column if and only if:

1) $\sum_{k=j}^{n-1} t_{k,j-1} \neq 0$, and

2) $\sum_{k=j-1}^{n-1} t_{kl} \neq 0$.

Theorem 4.1 All possible paths between a given pair of nodes, say 1 and 2, in a graph can be listed through application of the following steps:

Step 1 Associate with the given graph the edge T-matrix $\lfloor T_1$, using Definition 3.1-b if the graph is non oriented, and Definition 3.1-c if the graph is oriented.

Step 2 Generate the 2-subset of subsequent edge T-matrices derived from $\lfloor T_1$, using Property 3.14, provided that each column to which the C-transformation is applied must be a P-operating column. If in any newly generated edge T-matrix, say $\lfloor T_{mj}$, the entry $t_{j-1,j-1}$ is zero, then repeat the C-transformation with respect to the same column of $\lfloor T_{mj}$, if it is a P-operating column, to form $\lfloor T_{mj}{}_2$, then discard $\lfloor T_{mj}$.

Step 3 Let P_k represent a path of length k, i.e. a path composed of k edges connecting nodes 1 and 2. From each member of the 2-subset of subsequent edge T-matrices generated in Step 2, P_k is given by:

$P_k = t_{kl} \prod_{i=2}^{k} t_{ii}$, if the graph is nonoriented, and

(if the graph is oriented all the entries under the multiplication sign must be of the same sign which is opposite to that of t_{kl}) with $1 \leq k \leq n-1$, if P_k is computed from $\lfloor T_1$ and with $j-1 \leq k \leq n-1$ if P_k is computed from an edge T-matrix generated from its antecedent by applying the C-transformation corresponding to column $j \geq 3$.

Proof: From Property 3.6 it results that the expression of P_k given in Step 3 represents respectively a path of length k connecting node 1 and 2 in a nonoriented and an oriented graph. Since P_k's are computed from the 2-subset of subsequent edge T-matrices derived from the original ege T-matrix $\lfloor T_1$, from Property 3.1, it results that, given a value of k, with $2 \leq k$ n-2, all combinations of k-1 nodes which can be intermediate nodes in the path, taken out of n-2 nodes of the graph are all considered. Therefore all possible paths between nodes 1 and 2 are listed. The duplication is prevented by the restriction on the computation of P_k from $\lfloor T_{mj}$, since from the definition of the C-transformation, all entries in rows i, with $1 \leq i \leq j-1$, remain unchanged, therefore all paths of length $k < j-1$ have been listed, and then only paths of length $k \geq j-1$ are of interest. The restriction on the generation of subsequent edge T-matrices results from (a) the definition of the P-operating column and (b) the condition given in Step 2, is of purpose of reducing the generation of useless edge T-matrices from which no new path can be obtained.

Example 4.2 List all paths in the directed graph G_1 of Figure 4.2 between nodes 1 and 2.

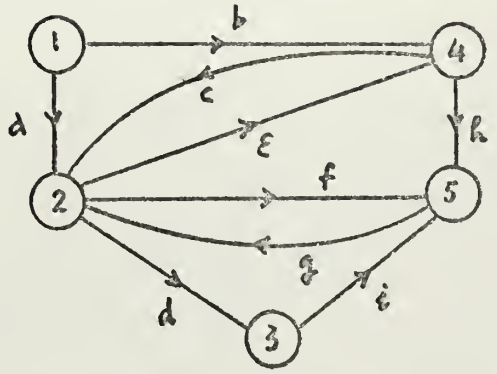


Figure 4.2. Illustration of the Path Listing Algorithm.

Step 1. The edge T-matrix associated with G_1 is:

$$\lfloor T_1 = \begin{vmatrix} a & & & \\ 0 & d & & \\ b & e-c & 0 & \\ 0 & f-g & i & h \end{vmatrix}$$

Step 2. The 2-subset of subsequent edge T-matrices derived from $\lfloor T_1$ are:

$$\lfloor T(13) = \begin{vmatrix} a & & & \\ b & e-c & & \\ 0 & f-g & h & \\ 0 & 0 & 0 & -1 \end{vmatrix} \qquad \lfloor T(14) = \begin{vmatrix} a & & & \\ 0 & d & & \\ 0 & f-g & i & \\ b & e-c & 0 & -h \end{vmatrix}$$

$\lfloor T(134)$ does not exist.

$$\underline{|T(13^2)} = \begin{vmatrix} a & & & \\ 0 & f-g & & \\ 0 & -d & -l & \\ b & e-c & -h & 0 \end{vmatrix} \qquad \underline{|T(13^24)} = \begin{vmatrix} a & & & \\ 0 & f-g & & \\ b & c-e & -h & \\ 0 & d & i & 0 \end{vmatrix}$$

Step 3. Listing oriented paths.

From $\underline{|T(1)}$: $P_1 = a$, From $\underline{|T(13)}$: $P_2 = bc$

From $\underline{|T(13^2)}$ and $\underline{|T(14)}$: none

From $\underline{|T(13^24)}$: $P_3 = bgh$

It is also interesting to note that, if the edge orientation is ignored, then the path matrix of the graph, now became nonoriented, is:

	a	b	c	d	e	f	g	h	i
	1	.	.	.	.	.	.	.	.
	.	1	1	.	.	.	.	.	.
	.	1	.	.	1	.	.	.	.
	.	1	.	.	.	1	.	1	.
	.	1	.	.	.	.	1	1	.
	.	1	.	1	.	.	.	1	1

Remark 4.1 Due to the fact that no duplication occurs in the listing of paths from each member of the 2-subset of subsequent edge T-matrices derived from T(1), then paths can be listed from each member of the set as soon as it is generated. Then, the maximum number of edge T-matrices in temporary storage, in the worst case, is only n-2. Hence this procedure enables one to reduce computer memory requirement and also computation time.

D. CIRCUIT LISTING ALGORITHM

In this section the path listing algorithm given in Section 4.3 will be modified to use for listing all circuits of a graph. The concept of primitive circuit is first introduced in order to cover the case of graphs with parallel edges. A partition of the set of all circuits of a graph will be proposed, from which an iterative method for listing all circuits of a graph will be pointed out.

1. Primitive Circuit.

Definition 4.6 A primitive circuit is a circuit formed by a pair of parallel edges if the graph is nonoriented. If the graph is oriented, a primitive circuit is formed by a pair of parallel edges whose directions are opposite.

Property 4.1 The number of nonoriented primitive circuits obtained from a set of P parallel edges is:

$$N = \tfrac{1}{2}P(P-1) \qquad (4.1)$$

Proof: From Definition 4.6, it is clear that the number of primitive circuits formed by P nonoriented edges is the number of all combination of P edges taken 2 edges at a time. Therefore

$$N = \binom{P}{2} = \tfrac{1}{2}P(P-1)$$

Property 4.2 Suppose that between a given pair of nodes, there are P_1 edges oriented in a direction and P_2 edges oriented in the opposite direction, then the number of primitive oriented circuits is:

$$N' = P_1 \cdot P_2 \qquad (4.2)$$

Proof: Consider one out of P_2 edges. This edge forms with every one edge of P_1 an oriented circuit. Hence property 4.2.

2. Iterative Process for Listing All Circuits of a Graph

Consider a given graph G, its vertices are numbered from 1 to n. Suppose that all of primitive circuits of G have been listed using Definition 4.2. Let C denote the set of all distinct circuits of the graph formed at least of three edges of the original graph. With respect to node 1, we can write

$$C = C(1) + C'(1) \qquad (4.3.a)$$

where C(1) is the subset of the set C whose member are circuits containing node 1 and C'(1) its complement. Since C'(1) is a subset of circuits of G that do not contain node 1, then C'(1) can be obtained from the subgraph G_1 of G obtained by deleting node 1 and all of the edges incident at this node. Repeating the process i times gives:

$$C'(i) = C(i+1) + C'(i+1) \qquad (4.3.b)$$

with successively i = 2,3,...,n-3, and where C(i+1) is the subset of circuits of G which do not contain the first i nodes of G but all contain node i+1, and C'(i+1) its complement in the set of circuits obtained from the graph G_i derived from G by deleting the first i nodes and all of the

edges incident at these nodes. In particular the subgraph G_{n-3} is formed by the last three nodes of G, then

$$C'(n-4) = C(n-3) \qquad (4.3.c)$$

Adding Equations 4.3 leads to

$$C = \sum_{i=1}^{n-3} C(i) \qquad (4.3.d)$$

Equation 4.3.d suggests that the set of all possible circuits of a graph G can be obtained by summing n-3 subsets C(i) which in turn can be determined by finding circuits that contain node i in the subgraph G_i.

3. The Modified C-transformation

In order to derive a computational iterative method for listing all circuits of a graph using the edge T-matrix and the process given in Section 4.4.2, we are going to modify the C-transformation as follows.

Definition 4.7 Any column j of an edge T-matrix is a C'-operating column if, for $j > 2$:

$$\sum_{k=j}^{n-1} t_{k,j-1} \neq 0, \text{ and}$$

$$\sum_{k=j-1}^{n-1} t_{k,1} \neq 0$$

In particular, for $j = 2$, column 2 of any edge T-matrix is a C'-operating column if at least there are two non zero entries in the set of elements in the first column from the second row down to the last row.

Definition 4.8 Given the edge T-matrix $\lfloor T_m = \{t_{pq}\}$ associated with a graph G_m. It is said that the edge T-matrix $\lfloor T_{mj} = \{t'_{pq}\}$ is derived from $\lfloor T_m$ by applying the modified C'-transformation using column $j > 2$ as C'-operating column if:

(1) $t'_{pq} = t_{pq}$ for $1 \leq p,q \leq j-2$

(2) $t'_{pq} = t_{p+1,q}$ for $j-1 \leq p \leq n-2$ and $1 \leq q \leq j-1$

(3) $t'_{pq} = t_{p+1,q+1}$ for $j \leq p \leq n-2$ and $j \leq q \leq n-2$

(4) $$t'_{n-1,q} = \begin{cases} t_{j-1,q} & \text{for } 1 \leq q \leq j-1 \\ t_{qj} & \text{for } j \leq q \leq n-1 \end{cases}$$

In particular for $j = 2$, then:

(5) $t'_{i,1} = t_{i+1,1}$ for $1 \leq i \leq n-2$

(6) $t'_{n-1,1} = 0$

(7) $t'_{pq} = t_{p+1,q+1}$ for $2 < p,q < n-2$

(8) $t'_{n-1,q} = t_{q,2}$ for $2 < q < n-1$.

Compared to the definition of the C-transformation, the modified C'-transformation is identical to the former except for:

(a) Only columns of $\lfloor T_m$ which satisfy conditions given in Definition 4.7 can be used as operating column, and

(b) The entry t'_{pq} is replaced by a zero instead of $t_{j-1,1}$.

4. Circuit Listing Theorem

Theorem 4.2 The set of all circuit of a graph can be obtained through application of the following steps.

Step 1. Associate with the graph the edge T-matrix $\lfloor T_1$. Using Definition 3.1-b for nonoriented graph and Definition 3.1-c for oriented graph. List the set of primitive circuits.

Step 2. Generate the 1-subset of subsequent edge T-matrices from $\lfloor T_1$, using the C'-transformation. If in each newly generated edge T-matrix, say $\lfloor T_{mj}$, the entry $t_{j-1,j-1}$ is zero, then repeat the C'-transformation for $\lfloor T_{mj}$ with respect to the same column j, to obtain $\lfloor T_{mj2}$ and discard $\lfloor T_{mj}$.

Step 3. Delete column 1 from T(1) and repeat Step 2.

Step 4. Repeat Step 3 by deleting one more succeeding column each time, until n-3 columns are deleted.

Step 5. For each k, with $3 \leq k \leq n$, obtain a set of C_k of all k-edge circuits from each edge T-matrix generated in Steps 1 to 4, using the formula

1. For nonoriented graph

$$C_k = t_{k-1,1} \prod_{i=1}^{k-1} t_{ii}$$

with $3 \leq k \leq n-1$ for $\lfloor T_1$ and $\lfloor T_{12^r}$ where $1 \leq r \leq n-2$, and

$j \leq k \leq n-1$ for $\lfloor T_{mj}$, where $j > 2$.

2. For oriented graph, C_k represent a circuit product if all elements in teh product are of same sign and opposite to that of $t_{k-1,1}$.

Proof: Applying Property 3.6, it is clear that C_k is the circuit product of a k-edge circuit. The proof that

follows will consist of three parts. It will be shown successively that applying the circuit listing theorem the circuits of a graph can be obtained from the set of C'-subsequent T-matrices without duplication.

The restriction outlined in Step 2, namely discarding $\lfloor\underline{T_{ij}}$, if its element $t_{j-1,j-1} = 0$, prevents the generation of subsequent edge T-matrices from which all C_k computed are zero. This results from the fact that $t_{j-1,j-1}$ remains zero in all of the subsequent edge T-matrices derived from $\lfloor\underline{T_{1j}}$, except for $\lfloor\underline{T_{1j}}^2$ those derived from $\lfloor\underline{T_{1j}}^2$. It is also the purpose of the choice of the operating column as defined in Definition 4.7. Suppose that an arbitrary column j has been used as operating column and if conditions given in Definition 4.7 are not satisfied, then it is easy to verify that the expression of C_k computed from the resulting edge T-matrix will be zero.

Consider the set of edge T-matrices generated in Step 2. The second column of these edge T-matrices are successively associated with nodes i, with $2 \leq i \leq n$. Therefore C_k's computed from these edge T-matrices form the complete set of C(1), i.e. the subset of circuits that contain node 1. Next, consider the set of edge T-matrices generated in Steps 3 and 4. These edge T-matrices are associated with subgraph G_i, derived from G by deleting the first i nodes and all of the edges incident at these nodes. Since Step 2 is repeated with respect to these newly formed edge T-matrices, then all subset C(i) of circuits of the original

graph will be successively computed. Applying the iterative process of circuit listing discussed in Section 4.4.2, shows that all of the circuits of the given graph are listed.

Finally, the prevention of duplication in listing these circuits is performed by (a) the application of the definition of the C'-transformation, namely letting $t_{n-1,1} = 0$ in the generation of $\underline{|T_{12^r}}$, for $1 \leq r \leq n-2$, and (b) the restriction on the computation of C_k, given in Step 5, from $\underline{|T_{mj}}$, where $j > 2$. The prevention of duplication can be explained by noting that the process of generating a new edge T-matrix results from a circular substitution of order j-1 if the j^{th} column is used as operating column. When $j = 2$, this operation conserves the n-edge circuit, then to prevent duplication, $t_{n-1,1}$ in the newly generated edge T-matrix, $\underline{|T_{12^r}}$, must be set equal zero to eliminate this n-edge circuit which has been listed from the antecedent edge T-matrix of $\underline{|T_{12^r}}$. For $j > 2$, since the circular substitution affects only the n-j last elements in the node sequence, then only circuits of length $k \geq j$ are newly generated, all circuits of length $k < j$ have been listed previously. Therefore only C_k with $j \leq k \leq n$ are to be computed from $\underline{|T_{mj}}$. Hence Theorem 4.2 is proved.

Example 4.3 List all circuits in the oriented graph of Figure 4.3.

Step 1. The edge T-matrix associated with G is

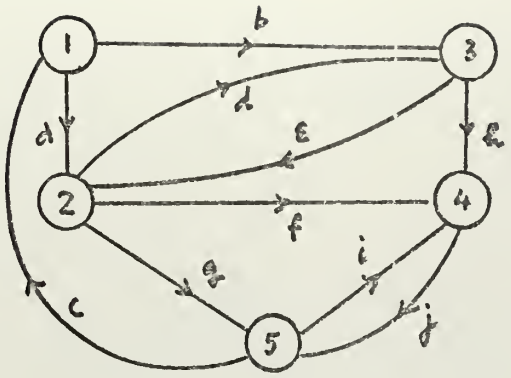


Figure 4.3. Illustration of Circuit Listing Algorithm.

$$\left|\underline{T_1}\right| = \left|\begin{array}{cccc} a & & & \\ b & d\text{-}e & & \\ . & f & h & \\ -c & g & . & j\text{-}l \end{array}\right|$$

The primitive circuits of G are:

$C_2 = de$

$C_2 = ij$

Step 2. The 1-subset of subsequent edge T-matrices derived from $\left|\underline{T_1}\right|$ is generated below.

$$\left|\underline{T_{12}}\right| = \left|\begin{array}{cccc} b & & & \\ . & h & & \\ -c & . & j\text{-}l & \\ . & e\text{-}d & -f & -g \end{array}\right| \qquad \left|\underline{T_{13}}\right| = \left|\begin{array}{ccc} a & & \\ . & f & \\ -c & g & j\text{-}l \\ -b & e\text{-}d & -h \end{array}\right| .$$

$\left|\underline{T_{14}}\right|$ does not exist, since column 4 in $\left|\underline{T_1}\right|$ is not a C'-operating column.

$$\underline{|T_{123}} = \begin{vmatrix} b & & & \\ -c & & & \\ . & e-d & -g & \\ . & -h & i-j & f \end{vmatrix} \qquad \underline{|T_{123^2}} = \begin{vmatrix} b & & & \\ . & e-d & & \\ . & -h & f & \\ -c & . & g & j-l \end{vmatrix}$$

Since the entry t_{22} in $\underline{|T_{123}}$ is zero, then it is to be discarded after $\underline{|T_{123^2}}$ has been generated.

$$\underline{|T_{124}} = \begin{vmatrix} b & & & \\ . & h & & \\ . & e-d & -f & \\ c & . & i-j & g \end{vmatrix} \qquad \underline{|T_{13^2}} = \begin{vmatrix} a & & & \\ -c & g & & \\ -b & e-d & . & \\ . & -f & i-j & h \end{vmatrix}$$

$$\underline{|T_{134}} = \begin{vmatrix} a & & & \\ . & f & & \\ -b & e-d & -h & \\ c & -g & i-j & \end{vmatrix} \qquad \underline{|T_{13^24}} = \begin{vmatrix} a & & & \\ -c & g & & \\ . & -f & i-j & \\ b & d-e & . & -h \end{vmatrix}$$

$$\underline{|T_{123^24}} = \begin{vmatrix} b & & & \\ . & e-d & & \\ -c & . & g & \\ . & h & -f & i-j \end{vmatrix}$$

Step 3 and Step 4.

$$\underline{|T_1^1} = \begin{vmatrix} d-e & & \\ f & h & \\ g & . & j-l \end{vmatrix} \qquad \underline{|T_{12}^1} = \begin{vmatrix} f & & \\ g & j-l & \\ . & -h & . \end{vmatrix}$$

$$\left|\underline{T^1_{123}}\right. \quad \begin{array}{|ccc} f & & \\ . & -h & \\ +g & j-1 & . \end{array} \qquad \left|\underline{T^2_1}\right. = \begin{array}{|cc} h & \\ . & j-1 \end{array}$$

Step 5. Computing C_k

From $\left|\underline{T_1}\right.$: C_5 = cadjh
From $\left|\underline{T_{13}}\right.$: C_4 = cafg
From $\left|\underline{T_{124}}\right.$: none
From $\left|\underline{T_{134}}\right.$: none
From $\left|\underline{T_{123^24}}\right.$: C_4 = cbeg
From $\left|\underline{T^1_{12}}\right.$: none
From $\left|\underline{T^2_1}\right.$: none.

From $\left|\underline{T_{12}}\right.$: C_4 = cbhj
From $\left|\underline{T_{123^2}}\right.$: C_5 = cbefj
From $\left|\underline{T_{13^2}}\right.$: C_3 = cag
From $\left|\underline{T_{13^24}}\right.$: none
From $\left|\underline{T^1_1}\right.$: none
From $\left|\underline{T^1_{123}}\right.$: none

Remark 4.2 Due to the fact that no duplication exists in the listing of circuits from each member of the set of subsequent edge T-matrices generated in Steps 2, 3 and 4, circuits can be listed from each edge T-matrix as soon as it is generated. Therefore, the maximum number of edge T-matrices in temporary storage, in general, is n-1. Hence this method enables one to reduce the memory requirement and computation time of the computer.

E. SEG LISTING ALGORITHM

The purpose of this section is to derive an algorithm for listing, without duplication, all of the $2^{n-1} - 1$ non empty segs of a graph of n vertices, by use of the edge T-matrix and the C-transformation.

1. Determination of a Seg from an Edge T-matrix

Property 4.3 Given a value of K, with $1 \leq K \leq n-1$, a seg S_K, determined by partitioning the set of nodes of the graph into two subsets, N_1 and N_2 with K nodes in N_1 and the remaining n-K nodes in N_2, can be obtained by taking all non zero entries t_{ij} with $K \leq i \leq n-1$ and $1 \leq j \leq K$, in the edge T-matrix associated with the graph.

Proof: Replacing all non zero entries t_{ij} for $K \leq i \leq n-1$ and $1 \leq j \leq K$ by zero in the edge T-matrix associated with the given graph, the resulting edge T-matrix will be associated with a disconnected graph by virtue of Property 3.4. Then the set S_k formed by non zero entries t_{ij} defined above is clearly a seg of the graph by Definition 2.9. In particular, if the subgraphs associated with the sub-edge T-matrices formed by considering the first k-1 rows and the last n-k-1 columns of the original edge T-matrix, respectively, are not disconnected then S_K is a minimal cutset of in short, a cutset.

2. The S-subset of Subsequent Edge T-matrices

Property 4.3 suggests that the set of all of the $2^{n-1} - 1$ segs of a graph of n vertices can be obtained from the complete set of subsequent edge T-matrices derived from the original edge T-matrix of the graph. A further investigation shows that without a systematic method, the segs computed from every member of the complete set of subsequent edge T-matrix by using Property 4.3 will be duplicated. In addition the set of segs can be listed from only a subset

of the set of subsequent edge T-matrices of the original edge T-matrix. Therefore to prevent duplication and at the same time to reduce computation time the concept of S-operating column and the set of S-subset of subsequent edge T-matrices will now be introduced.

Definition 4.9 Definition of the S-operating column.

a) Given the edge T-matrix $\lfloor \underline{T}$ associated with a graph of n vertices. Any column j of $\lfloor \underline{T}$ is called an S-operating column if:

$$1 \leq j \leq j_m \quad \text{with} \quad j_m = \begin{cases} \frac{1}{2}(n-1) \text{ for } n \text{ odd}, \\ \frac{1}{2}n \text{ for } n \text{ even}. \end{cases}$$

b) More generally, let $T(j^{\alpha},k^{\beta},\ldots,m^{\mu})$ be the edge T-matrix derived from T by applying the C-transformation with respect to the j^{th} column α times, the k^{th} column β times,... the m column μ times, then the p column with $p \geq m$ is a S-operating column if:

$$(\alpha + \beta + \ldots + \mu) \leq n-p. \tag{4.4}$$

Definition 4.10 Given an edge T-matrix $\lfloor \underline{T}$, associated with a graph, the S-subset of subsequent edge T-matrices derived from $\lfloor \underline{T}$ is the set of edge T-matrix obtained as follows:
1) Applying the C-transformation with respect to every S-operating column of $\lfloor \underline{T}$,
2) From each newly generated edge T-matrix $\lfloor \underline{T_j}$ obtained in Step 1, obtain a set of edge T-matrices $\lfloor \underline{T_{jk}}$, provided that column k is an S-operating column.

3) Repeat Step 2 until no column in the newly generated edge T-matrices can be used as S-operating column.

Some interesting properties of the S-subset of subsequent edge T-matrices can be derived from Definitions 4.9 and 4.10. These properties will lead to a seg listing algorithm by which all segs of a graph can be listed without duplication from the S-subset of subsequent matrices of the graph.

Definition 4.11 Let $\lfloor T_{j^{\alpha},k^{\beta},\ldots,m^{\mu}}$ be a member of the S-subset of subsequent edge T-matrices associated with a graph. $\lfloor T_{j^{\alpha},k^{\beta},\ldots,m^{\mu}}$ is called a member of the g^{th} generation, if the number of times that the C-transformation has been used to derive this edge T-matrix is exactly g, in other words

$$\alpha + \beta + \gamma + \ldots + \mu = g.$$

Property 4.4 The S-subset of subsequent edge T-matrices derived from the original edge T-matrix T, can be partitioned into n-1 generations.

Proof: From Eq. 4.4, let p = 1, then $\beta, \gamma, \delta \ldots, \mu = 0$ and then $\alpha = n-1$. Note that the original edge T-matrix T is considered as belonging to the generation g = 0.

Definition 4.12 Given an integer K, with $1 \leq K \leq j_m$. A member of the generation g is said to belong to a family K if it has been derived from $\lfloor T$ by using the C-transformation exactly g times with respect only to column j such that $1 \leq j \leq K$.

Property 4.5 The number of members of a family K, in generation g is:

$$f(K,g) = \binom{K+g-1}{g} \tag{4.5}$$

Proof: From Definition 4.12, a member of the family K is derived from the original matrix by applying the C-transformation exactly g times with respect to the first K columns, then the number of members of family K in the g^{th} generation is clearly the number of g-combinations of K distinct objects, each of which may appear 0 to g times. It is well known that this number is $\binom{K=g-1}{g}$.

Property 4.6 Members of a family K exist only in the first n-K generations.

Proof: From Eq. 4.4

$$g = \alpha + \beta + \ldots + \mu \leq n-p$$

Then, let p = K,

$$g \leq n-K$$

Hence Property 4.6.

Property 4.7 The total number of members of family K in the S-subset of subsequent edge T-matrices derived from the original edge T-matrix is:

$$N(K) = \binom{n}{K} \tag{4.6}$$

Proof: It has been shown that:

$$\binom{n}{r} = \binom{n-1}{r} + \binom{n-2}{r-1} + \ldots + \binom{n-1-r}{0} \tag{4.7}$$

Now, the total number of members of family K in the S-subset of subsequent edge T-matrices derived from the original edge T-matrix is:

$$N(K) = \sum_{g=0}^{n-K} \binom{K+g-1}{g} \qquad (4.7.a)$$

Letting r = n - K, and substitute into Eq. 4.7.a, gives

$$N(K) = \binom{n-r-1}{0} + \binom{n-r}{1} + \binom{n-r+1}{2} + \ldots + \binom{n-2}{r-1} + \binom{n-1}{r} \qquad (4.7.b)$$

Comparing Eq. 4.7.b with Eq. 4.7, shows that

$$N(K) = \binom{n}{n-K} = \binom{n}{K}$$

Hence Eq. 4.6 is proved.

Property 4.8 Let $s_m(K)$ be the sequence of nodes associated with the first K columns of an arbitrary member $\lfloor T_m$ of a family K in the S-subset of subsequent edge T-matrices of a graph G. Then the set composed of $\binom{n}{K}$ such sequences obtained from every member of the family K form the whole combinations of n nodes of G taken K nodes at a time.

Proof: Property 4.8 results directly from the fact that the number of members of the family K is exactly $\binom{n}{K}$, and since only K first columns in each member are considered, with $1 \leq K \leq j_m$ and from Definition 4.12, members of family K are obtained by applying the C-transformation with respect to column j, with $1 \leq j \leq K$, then in the light of Remark 3.2, all $s_m(K)$ are distinct. Hence Property 4.8 is proved.

The foregoing discussion suggests the following seg listing theorem.

3. Seg Listing Theorem

Theorem 4.3 All of the $2^{n-1} - 1$ non empty segs of a graph G of n vertices can be listed without duplication through application of the following steps.

Step 1. Associate with G its edge T-matrix $\lfloor T$.

Step 2. Generate from $\lfloor T$ its S-subset of subsequent edge T-matrices. Using Definitions 4.9 and 4.10.

Step 3. For each value of K, using Property 4.3, list S_K, with:

$$1 \leq K \leq j_m, \text{ if } S_K \text{ is determined from } \lfloor T,$$

and

$$m \leq K \leq p, \text{ if } S_K \text{ is determined from } \left\lfloor T_{j^\alpha, k^\beta, \ldots, m^\mu} \right.$$

where

$$p = \begin{cases} j_m \text{ when } g = \alpha + \beta + \ldots + \mu \geq n - j_m \\ n - g, \text{ otherwise.} \end{cases}$$

Proof: Suppose that S_K is computed from every member of the family K, then $\binom{n}{K}$ segs will be obtained. None of these segs can be empty, since it is assumed that the given graph is non separable, and since the graph associated with any edge T-matrix, from which S_K is computed, is 2-isomorphic with the given graph, then if S_K is empty the latter is a separable graph. This is impossible. Now let K be $1,2,\ldots, j_m$, then the total number of segs obtained will be:

$$S_t = \sum_{K=1}^{j_m} \binom{n}{K} = 2^{n-1} - 1.$$

All of the segs listed are clearly distinct by virtue of Property 4.8.

The practical rules given in Step 3 result from Property 4.6. The original edge T-matrix T can be considered as member of any family K, with $1 \leq K \leq j_m$. Members of family $K = j_m$ can only belong to the first j_m generations while those of family $K \geq j_m$ belong to generations $0 \leq g \leq n-K$.

Example 4.4 List all segs of the graph G of Figure 4.4.

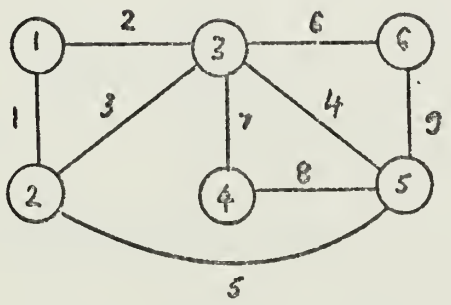


Figure 4.4. Illustration of the Seg Listing Algorithm

Step 1. The edge T-matrix associated with G is

$$\underline{T} = \begin{vmatrix} 1 & & & & \\ 2 & 3 & & & \\ 0 & 0 & 7 & & \\ 0 & 5 & 4 & 8 & \\ 0 & 0 & 6 & 0 & 9 \end{vmatrix}$$

Step 2. Generation of the S-subset of subsequent edge T-matrices of T. Since n = 6, then:

$$j_m = \tfrac{1}{2}n = 3.$$

Members of the first generation are:

$$|\underline{T(1)} = \begin{vmatrix} 3 & & & & \\ 0 & 7 & & & \\ 5 & 4 & 8 & & \\ 0 & 6 & 0 & 9 & \\ 1 & 2 & 0 & 0 & 0 \end{vmatrix} \quad |\underline{T(2)} = \begin{vmatrix} 2 & & & & \\ 0 & 7 & & & \\ 0 & 4 & 8 & & \\ 0 & 6 & 0 & 9 & \\ 1 & 3 & 0 & 5 & 0 \end{vmatrix} \quad |\underline{T(3)} = \begin{vmatrix} 1 & & & & \\ 0 & 0 & & & \\ 0 & 5 & 8 & & \\ 0 & 0 & 0 & 9 & \\ 2 & 3 & 7 & 4 & 6 \end{vmatrix}$$

Member of the second generation are:

$$|\underline{T(1^2)} = \begin{vmatrix} 7 & & & & \\ 4 & 8 & & & \\ 6 & 0 & 9 & & \\ 2 & 0 & 0 & 0 & \\ 3 & 0 & 5 & 0 & 1 \end{vmatrix} \quad |\underline{T(12)} = \begin{vmatrix} 0 & & & & \\ 5 & 8 & & & \\ 0 & 0 & 9 & & \\ 1 & 0 & 0 & 0 & \\ 3 & 7 & 4 & 6 & 2 \end{vmatrix} \quad |\underline{T(2^2)} = \begin{vmatrix} 0 & & & & \\ 0 & 8 & & & \\ 0 & 0 & 9 & & \\ 1 & 0 & 5 & 0 & \\ 2 & 7 & 4 & 6 & 3 \end{vmatrix}$$

$$|\underline{T(23)} = \begin{vmatrix} 2 & & & & \\ 0 & 4 & & & \\ 0 & 6 & 9 & & \\ 1 & 3 & 5 & 0 & \\ 0 & 7 & 8 & 0 & 0 \end{vmatrix} \quad |\underline{T(3^2)} = \begin{vmatrix} 1 & & & & \\ 0 & 5 & & & \\ 0 & 0 & 9 & & \\ 2 & 3 & 4 & 6 & \\ 0 & 0 & 8 & 0 & 7 \end{vmatrix}$$

Members of the third generation are:

$$|\underline{T(1^3)} \quad \begin{vmatrix} 8 & & & & \\ 0 & 9 & & & \\ 0 & 0 & 0 & & \\ 0 & 5 & 0 & 1 & \\ 7 & 4 & 6 & 2 & 3 \end{vmatrix} \quad |\underline{T(1^2 2)} = \begin{vmatrix} 4 & & & & \\ 6 & 8 & & & \\ 2 & 9 & 0 & & \\ 3 & 5 & 0 & 1 & \\ 7 & 8 & 0 & 0 & 0 \end{vmatrix} \quad |\underline{T(12^2)} = \begin{vmatrix} 5 & & & & \\ 0 & 9 & & & \\ 1 & 0 & 0 & & \\ 3 & 4 & 6 & 2 & \\ 0 & 8 & 0 & 0 & 7 \end{vmatrix}$$

$$|\underline{T(2^3)} = \begin{vmatrix} 0 & & & & \\ 0 & 9 & & & \\ 1 & 5 & 0 & & \\ 2 & 4 & 6 & 3 & \\ 0 & 8 & 0 & 0 & 7 \end{vmatrix} \quad |\underline{T(2^2 3)} = \begin{vmatrix} 0 & & & & \\ 0 & 0 & & & \\ 1 & 0 & 0 & & \\ 2 & 7 & 6 & 3 & \\ 0 & 8 & 9 & 5 & 4 \end{vmatrix} \quad |\underline{T(23^2)} = \begin{vmatrix} 2 & & & & \\ 0 & 6 & & & \\ 1 & 3 & 0 & & \\ 0 & 7 & 0 & 0 & \\ 0 & 4 & 9 & 5 & 8 \end{vmatrix}$$

$$|\underline{T(3^2)} = \begin{vmatrix} 1 & & & & \\ 0 & 0 & & & \\ 2 & 3 & 6 & & \\ 0 & 0 & 0 & 7 & \\ 0 & 5 & 9 & 4 & 8 \end{vmatrix}$$

Members of the fourth generation are:

$$\underline{|T(1^4)} = \begin{array}{|l} 9 \\ 0\ 0 \\ 5\ 0\ 1 \\ 4\ 6\ 2\ 3 \\ 8\ 0\ 0\ 0\ 7 \\ \hline \end{array} \qquad \underline{|T(1^3 2)} = \begin{array}{|l} 0 \\ 0\ 0 \\ 0\ 0\ 1 \\ 7\ 6\ 2\ 3 \\ 8\ 9\ 0\ 5\ 4 \\ \hline \end{array} \qquad \underline{|T(1^2 2^2)} = \begin{array}{|l} 6 \\ 2\ 0 \\ 3\ 0\ 1 \\ 7\ 0\ 0\ 0 \\ 4\ 9\ 0\ 5\ 8 \\ \hline \end{array}$$

$$\underline{|T(13^3)} \quad \begin{array}{|l} 0 \\ 1\ 0 \\ 3\ 6\ 2 \\ 0\ 0\ 0\ 7 \\ 5\ 9\ 0\ 4\ 8 \\ \hline \end{array}$$

Member of the fifth generation is:

$$\underline{|T(1^5)} = \begin{array}{|l} 0 \\ 0\ 1 \\ 6\ 2\ 3 \\ 0\ 0\ 0\ 7 \\ 9\ 0\ 5\ 4\ 8 \\ \hline \end{array}$$

Step 3. Listing the complete set of segs.

The subset of segs S_1, with K = 1 are obtained from the family K = 1, in generation g = 0 down to g = n-K = 6-1 = 5, namely from $\underline{|T}; \underline{|T(1)}, \underline{|T(1^2)}, \underline{|T(1^3)}, \underline{|T(1^4)}, \underline{|T(1^5)}$. The number of segs obtained is

$$N(S_1) = \binom{6}{1} = 6.$$

The subset of segs obtained with K = 2 are determined from the family K = 2, from generation g = 0, down to generation g = n-K = 4, explicitly from:

$\underline{|T}; \underline{|T(1)}, \underline{|T(2)}; \underline{|T(1^2)}, \underline{|T(12)}, \underline{|T(2^2)}; \underline{|T(1^3)}, \underline{|T(1^2 2)}, \underline{|T(12^2)}, \underline{|T(2^3)}; \underline{|T(1^4)}, \underline{|T(13^2)}, \underline{|T(1^2 2^2)}, \underline{|T(12^3)}, \underline{|T(2^4)}$. The number of segs obtained is

$$N(S_2) = \binom{6}{2} = 15$$

With K = 3, the segs S_3 are obtained from the family K = 3, from generation g = 0 to generation g = n-K = 3, namely:

$\underline{|T}; \underline{|T(2)}, \underline{|T(3)}; \underline{|T(2^2)}, \underline{|T(23)}, \underline{|T(3^2)}; \underline{|T(2^3)}, \underline{|T(2^2 3)}, \underline{|T(23^2)}, \underline{|T(3^3)}$. The number of segs obtained is:

$$N(S_3) = \tfrac{1}{2} \binom{6}{2} = 10.$$

The total number of segs listed is:

$$N = N(S_1) + N(S_2) + N(S_3) = 6 + 15 + 10 = 31 = 2^5 - 1.$$

The seg matrix obtained is:

1	2	3	4	5	6	7	8	9
1	1	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0
0	1	1	1	0	1	1	0	0
0	0	0	0	0	0	1	1	0
0	0	0	1	1	0	0	1	1
0	0	0	0	0	1	0	0	1
0	1	1	0	1	0	0	0	0
1	0	1	1	0	1	1	0	0
1	1	0	0	0	0	1	1	0
1	1	0	1	1	0	0	1	1
1	1	0	1	1	1	1	0	0
0	1	1	1	0	1	0	1	0
0	0	0	1	1	0	1	0	1
0	0	0	1	1	1	0	1	0
1	0	1	0	1	0	1	1	0
1	0	1	1	0	0	0	1	1
1	0	1	0	1	1	0	0	1
0	1	1	0	1	1	1	1	1
0	1	1	1	0	0	1	0	1
0	0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	0	0
1	0	1	1	0	1	0	1	0
1	1	0	1	1	0	1	0	1
1	1	0	1	1	1	0	1	0
1	1	0	0	0	1	0	0	1
0	1	1	0	1	0	1	1	0
0	1	1	1	0	0	0	1	1
0	1	1	0	1	1	0	0	1
1	0	1	0	1	1	1	1	1
1	0	1	1	0	0	1	0	1
1	1	0	0	0	1	1	1	1

Remark 4.3 Since there is no duplication in the listing of segs by applying Theorem 4.3, then segs can be listed from each member of the S-subset of subsequent edge T-matrices as soon as it is generated. Therefore the maximum number of edge T-matrices in temporary storage is n-1. This enables one to predict the computer memory required.

V. EVALUATION OF TREE AND K-TREE SUMMATIONS

A. INTRODUCTION

The generation of trees of a graph is the most challenging problems in network topology. Quite a large number of solutions have been proposed. It has been shown by Chen [9, 10] that many of the existing techniques on generation of trees of a graph described in the literature, seemingly so different in their appearance, are actually the variant of the Wang algebra formulation, and that other existing techniques can be formulated by matrix algebra. Each of these methods is designed to list the trees or to evaluate the tree summation of one graph at a time. Although the same algorithm can be used to evaluate the tree summations of different graphs associated with the same network, the time involved in repeating the tree summation calculation necessary for various terms in a topological formula often makes the topological method unappealing for computer implementation. Thus, it is desirable to develop a method which can evaluate various 2-tree summations together with the 1-tree summation simultaneously. That is the main purpose of this chapter.

In the next sections, the concept of foldant given by Nakagawa [21] will be modified in such a way that the 1-tree summation and various 2-tree summations can be obtained from a set of T-subsequent edge T-matrices of the original network. Algorithm for determining the 1-tree

summation of a graph will be given in Section 5.2. This algorithm is comparable with the Chang's algorithm [6]. Algorithm for evaluating the Type-I and Type-II 2-tree summation from the same set of T-subsequent edge T-matrices will be outlined in Section 5.3 and Section 5.4, respectively. The advantages of the method are that it is easy to combine all of these algorithms together in such a way that the 1-tree summation and various 2-tree summations can be obtained at the same time. This is possible because all these 1-tree and 2-tree summations are computed from the same set of T-subsequent edge T-matrices, partially. The above mentioned algorithms are all designed for covering the case of directed graphs. This will be performed by using the *multiplication.

B. THE T-TRANSFORMATION AND THE SET OF SUBSEQUENT EDGE T-MATRICES

In order to use the edge T-matrix for the tree summation calculation we first modify the C-transformation defined in Section 3.4. The modified transformation is called the T-transformation.

Definition 5.1 Given an edge T-matrix $\lfloor T_1 = \{t_{pq}\}$ associated with a graph G of n vertices. It is said that the edge T-matrix $\lfloor T_{1j} = \{t'_{pq}\}$ is derived from $\lfloor T_1$ by applying the T-transformation using column j as operating column, if

(1) $t'_{pq} = t_{pq}$ for $1 \leq p,q \leq j-2$

(2) $t'_{pq} = t_{p+1,q}$ for $j-1 \leq p \leq n-2$, and $1 \leq q \leq j-1$

(3) $t'_{pq} = t_{p+1,q+1}$ for $j \leq p \leq n-2$, and $j \leq q \leq n-2$

(4) $t'_{n-1,q} = t_{qj}$ for $j \leq q \leq n-1$, and

$t'_{n-1,q} = 0$ for $1 \leq q \leq j-1$.

Definition 5.2 Given an edge T-matrix $\lfloor T_1$, associated with a graph of n nodes. The set of T-subsequent edge T-matrices derived from $\lfloor T_1$ is the set of edge T-matrices, which are individually associated with a connected graph, obtained by the following procedure:

(1) Apply the T-transformation with respect to column j of $\lfloor T_1$, for $2 \leq j \leq n-1$, then

(2) From each newly formed edge T-matrix, $\lfloor T_{1j}$, obtain a set of edge T-matrices $\lfloor T_{1jk}$ by applying the T-transformation relatively to column k, with $j \leq k \leq n-1$, and

(3) Repeat (2) on each newly generated edge T-matrix to obtain successively all members of the set.

Remark 5.1 Let G_1 be the graph whose edge T-matrix is $\lfloor T_1$. Let G_{1j} be the graph associated with the edge T-matrix $\lfloor T_{1j}$, derived from $\lfloor T_1$ by applying the T-transformation with respect to column j. From Definitions 3.1 and 5.1, it results that G_{1j} can be derived from G_1 by:

1. deleting all edges between node i+1 and nodes j, with $j \leq i+1$, then
2. rearranging vertices of G_1 in a sequences as follows:

   1,2,3,...,j-1,j+1,j+2,...,n-1,n,j.

Remark 5.2 From D finition 5.2, it follows that a column k can be used as operating column for generating a subsequent edge T-matrix if:

1. All entries in this column are not zero, and
2. All entries t_{pq}, with $1 \leq q \leq k-1$ and $k \leq p \leq n-1$ are not all zero, otherwise the graph G_k obtained will be a disjoint graph.

C. EVALUATION OF 1-TREE SUMMATION

To compute the 1-tree summation of a graph, the concept of the modified foldant is now introduced.

Definition 5.3 Let $\lfloor T_1$ be the edge T-matrix associated with a graph of n vertices. The modified foldant of $\lfloor T_1$ is defined by the following recursive rule

$$|T_1| = \sum_{k=1}^{n-1} t_{k,1} \, |T_{(2)}{}^{k-1}| \qquad (5.1)$$

where for k = 0, $|T_1'|$ is the modified foldant of $\lfloor T_1'$ which is the edge T-matrix derived from $\lfloor T_1$ be deleting the entry t_{11} and adding entries in the same row of column 1 and column 2; and $|T_{(2)}'{}^{k}|$ is obtained similarly from $\lfloor T_{(2)}{}^{k}$, which in turn is derived from $\lfloor T_1$ by applying successively k times the T-transformation using column 2 as operating column.

The following theorem has been proved by Percival [25]

Theorem 5.1 Let a graph be given with the tree summation $|T|$, then with respect to any branch designation b

$$|T| = |T'_o| + b|T_s|$$

where $|T'_o|$ is the tree summation of the graph derived from the given graph by deleting branch b, and $|T_s|$ is the tree summation of the graph obtained from the given graph by short-circuiting branch b.

With Theorem 5.1, Nagakawa [21] has proved the following corollary.

Corollary 5.1 Let a graph be given with the tree summation T. Then with respect to b_{ij}, the sum of all edge designations between vertices i and j, the tree summation is given by

$$|T| = |T'_d| + b_{ij} |T_s|$$

where $|T_o|$ is the tree summation of the graph derived from the given graph by deleting all edges between nodes i and j, and $|T_s|$ is the tree summation of the graph obtained from the given graph by short-circuiting nodes i and j.

Using Corollary 5.1, the following theorem concerning the evaluation of the tree summation of a graph will be proved. The case of the nondirected graph will be considered first.

Theorem 5.2 The tree summation of a nonoriented graph is equal to the modified foldant $|T_1|$ of the edge T-matrix $|T_1$ associated with the graph.

Proof: The proof will be achieved by induction. When the number of nodes of the graph is 2, the theorem is

obvious. Assume that the theorem is true when the number of nodes of the graph is n-1. Consider now a graph of n nodes associated with the edge T-matrix $\underline{|T_1}$, let $|T|$ denote the tree summation of the graph. Applying Corollary 5.1 with respect to t_{11}, the sum of edge sesignations between nodes 2 and 1, then:

$$|T| = |T_o| + t_{11} |T_s|.$$

Note that T_s is the tree summation of a graph of n-1 nodes, and it is verified that its edge T-matrix, $\underline{|T_1'}$, can be obtained from $\underline{|T_1}$ by deleting t_{11} and adding entries in the same row of column 1 and column 2. Also, since the graph associated with $\underline{|T_i'}$ is of n-1 nodes, then the induction hypothesis can be applied to $|T_i'|$. Next consider the tree summation T_o, of the graph derived from the given graph by deleting all edges between nodes 1 and 2. The tree summation is obviously independent of the nodes arrangement, then the nodes of this newly obtained graph are now rearranged in such a way that node i of the given graph is located in the $(i-1)^{th}$ location in the sequence, for $2 \leq i \leq n-1$, and node 1 of the given graph becomes the n^{th} element in the sequence. The edge T-matrix associated with this new graph is then the subsequent edge T-matrix derived from $\underline{|T_1}$ by using column 2 as operating column. Let $\underline{|T_{12}}$ be the resulting edge T-matrix, and $\underline{|T_{12}}$ its tree summation. Then

$$|T| = |T_{12}| + t_{11}|T_1'| \qquad (5.2)$$

Repeating the process with respect to $|\underline{T_{12}}$, we obtain

$$|T_{12}| = |T_{1(2)^2}| + t_{21}|T'_{12}| \tag{5.3}$$

By repeating the procedure n-1 times, then adding all the resulting equations, we have the theorem.

In order to derive a fast method for expanding the modified foldant, the following theorem will be proved.

Theorem 5.3 The tree summation of a graph is equal to the sum of partial tree summations obtained from every member of the set of subsequent edge T-matrices derived from the edge T-matrix $|\underline{T_1}$ associated with the original graph.

Proof: The proof will also be performed by induction. When the number of nodes is 2, the theorem is obvious. It is easy to verify for the case of a three nodes graph. Assume that the theorem is true when the number of nodes of the graph is n-1. Now consider a graph of n nodes whose tree summation is T. By Theorem 2

$$|T| = \sum_{k=1}^{n-1} t_{k,1} \left|T_{(2)^{k-1}}\right|. \tag{5.4}$$

Since $|T'_{(2)^k}|$ is the tree summation of a graph of n-1 vertices, then the induction hypothesis can be applied, thus $|T'_{(2)^k}|$ is equal to the sum of partial tree summations obtained from every member of the set of subsequent edge T-matrices derived from $|\underline{T'_{(2)^k}}$. Now consider a term in the right hand side of Equation 5.4, say $t_{k,1}|T'_{(2)^k}|$, corresponding to a given value of k. This product can be expanded

into a sum, of which each term corresponds to the product of $t_{k,1}$ and a partial tree summation obtained from an edge T-matrix, say $\left|\underline{T'_{(2)^k,j}}\right.$, which in turn is derived from $\left|\underline{T'_{(2)^k}}\right.$ by applying the T-transformation with respect to column $j \geq 2$. Let $\left|\underline{T_{(2)^k,j+1}}\right.$ be the edge T-matrix derived from $\left|\underline{T_{(2)^k}}\right.$ using column j+1 as operating column. By inspection it is verified that

$$\left|T_{(2)^k,j+1}\right|_p = \left|T'_{(2)^k,j}\right|_p \; t_{k+1,1}$$

for every value of j, with $j \geq 2$. Therefore, each term in the right hand side of Equation 5.4 can be obtained by the sum of all partial tree summations evaluated from every member of the set of subsequent edge T-matrices derived from $\left|\underline{T_{(2)^k}}\right.$ using column $j \geq 3$ as operating column. The total sum is then the sum of partial tree summation obtained from every member of the set of subsequent edge T-matrix derived from the original edge T-matrix $\left|\underline{T_1}\right.$. Thus Theorem 5.3 is proved.

Remark 5.3 From Definition 3.2, it is seen that the partial tree summation obtained from an edge T-matrix is zero if any row in this edge T-matrix contains all zero entries. Let $\left|\underline{T_{mj}}\right.$ be an edge T-matrix of which all entries in row j-1 are all zero. Suppose that all columns $k > j$ of this edge T-matrix can be used as operating column. If the T-transformation is applied with respect to column k to generate the subsequent edge T-matrix $\left|\underline{T_{mjk}}\right.$, from Definition 5.1 it follows that for all value of $k > j$, the row j-1

remains unchanged, then the partial tree summation evaluated from $\lfloor T_{mjk}$ is zero. Therefore the following rule can be used to avoid the generation of unuseful edge T-matrices.

Rule 5.1 If in a newly generated edge T-matrix $\lfloor T_{mj}$ obtained from $\lfloor T_{m}$ by applying the T-transformation with respect to column j, the entries in row j-1 are all zero, then repeat the T-transformation with respect to column j, if possible, to obtain $\lfloor T_{mj^2}$, and discard $\lfloor T_{mj}$.

Corollary 5.2 Theorem 5.3 can be applied to an oriented graph if Definition 3.1-a is used for describing the graph and if Definition 3.4 is applied for evaluating the partial directed tree summation from each member of the set of subsequent edge T-matrices derived from the original edge T-matrix.

Proof: From Definition 2.30, a directed tree must be a ordinary tree, and by Theorem 5.3 and Definition 3.4 all non directed trees can be obtained but only directed trees are taken into account in the computation of the directed tree summation.

Theorem 5.3 and Corollary 5.2 lead directly to the following tree summation evaluation algorithm.

Algorithm for computing 1-tree summation

Step I. Obtain from the original graph the associated edge T-matrix $\lfloor T_{1}$. Apply Definition 3.1-a for directed graph, and Definition 3.1-c for non directed graphs.

Step II. Generate the set of subsequent edge T-matrices from $\lfloor T_1$ by using the T-transformation. Apply Rule 5.1 if necessary. From each member of the set of subsequent edge T-matrices, evaluate the partial tree summation. Apply Definition 3.2 for non directed graph, and Definition 3.4 for directed graph.

Step III. Evaluate the tree summation of the graph by summing all partial tree summations obtained in Step II.

The following example illustrates the application of Theorem 5.3 and the algorithm.

Example 5.1 Evaluate the tree summation of the directed graph shown in Figure 5.1-a.

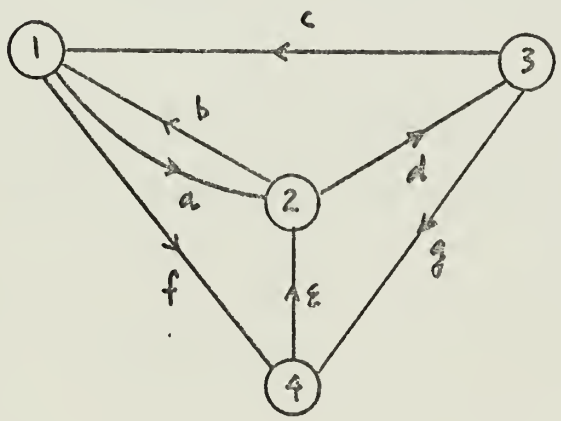


Figure 5.1.a. Illustration of 1-tree Summation Evaluation.

Step 1. The edge T-matrix associated with the graph is

$$\underline{|T_1|} = \begin{array}{c|ccc} 1 & (1,f) & & \\ 2 & 0 & (a,b)+(1,a) & \\ 3 & (3,g) & (3,c) & (2,d) \end{array}$$

The node 4 has been chosen as reference node, then edge e whose initial node is node 4 has been omitted in the edge T-matrix.

Step 2. The set of subsequent edge T-matrices derived from T_1 is:

$$\underline{|T_{12}|} = \begin{array}{|ccc|} \hline 0 & & \\ (3,g) & (2,d) & \\ 0 & (2,b)+(1,a) & (s,c) \\ \hline \end{array} \qquad \underline{|T_{1(2)^2}|} = \begin{array}{|ccc|} \hline (3,g) & & \\ 0 & (3,c) & \\ 0 & & (1,A)+(2,b) \\ \hline \end{array}$$

$$\underline{|T_{13}|} = \begin{array}{|ccc|} \hline (1,f) & & \\ (3,g) & (3,c) & \\ 0 & 0 & (2,d) \\ \hline \end{array} \qquad \underline{|T_{1(2)^2 3}|} = \begin{array}{|ccc|} \hline (3,g) & & \\ 0 & (2,d) & \\ 0 & 0 & (1,a)+(2,b) \\ \hline \end{array}$$

The partial directed tree summations obtained from every member of the set of subsequent edge T-matrices are:

$$|T_1|_p = fbg + fbc$$

$$|T_{13}|_p = fgd + fcd$$

$$|T_{1(2)^2 3}|_p = gda$$

The edge T-matrix $\underline{|T_{12}|}$ has been discarded by Rule 5.1.

Step 3. The directed tree summation of the graph is:

$$|T| = fbg + fbc + fgd + fcd + gda.$$

These directed trees are shown in Figure 5.1-b.

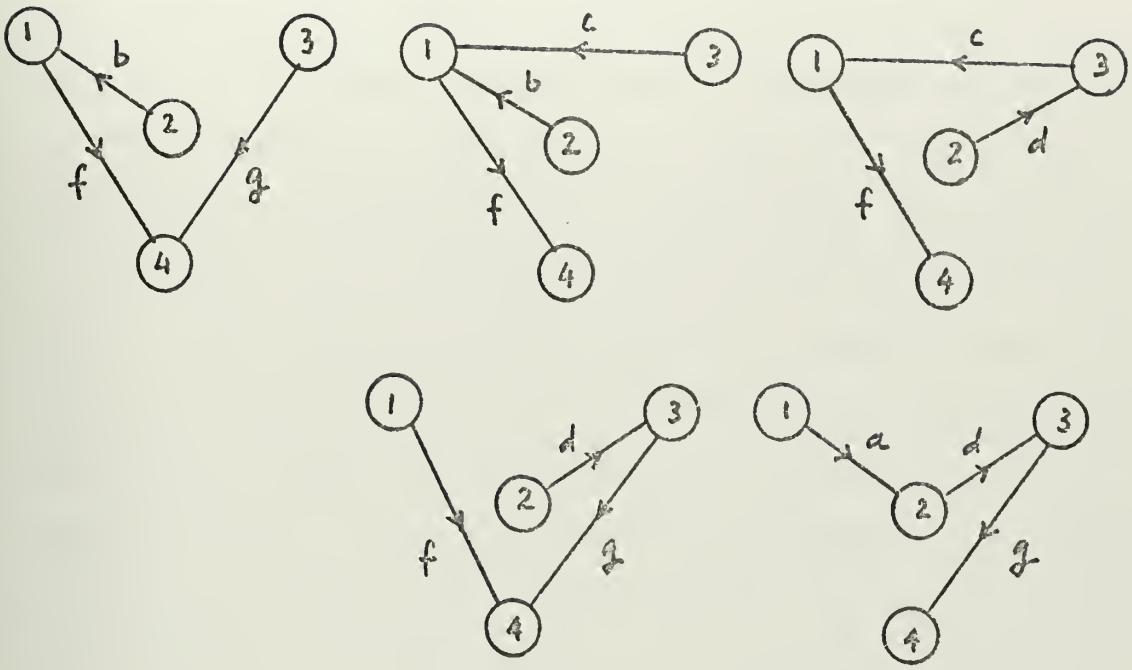


Figure 5.1.b. Directed 1-tree Obtained from the Graph of Figure 4.1.a.

D. EVALUATION OF TYPE-I 2-TREE SUMMATION

It has been proved [25] that, let $|T(a,r)|$ be the Type-I 2-tree summation of a graph G, then $|T(a,r)|$ is the 1-tree summation of the graph $G_{s(a,r)}$ derived from G by short-circuiting nodes a and r. Therefore $|T(a,r)|$ can be obtained by applying the 1-tree listing algorithm given in Section 5.3.1. to the graph $G_{s(a,r)}$. Unfortunately this simple procedure is unappealing for some applications. For instance, depending on the formula used, the evaluation of a driving-point function topologically requires two tree summations from two different network graphs [28], and if

an open circuit impedance matrix or a short-circuit admittance matrix are needed then five or seven 1-tree summations from five or seven distinct graphs [5, 12] are required. Therefore it is desirable to derive an algorithm for obtaining Type-I 2-tree summation at the same time the 1-tree summation is evaluated. That is the purpose of this section. In this order, it is observed that if the edge connecting nodes a and r in the original graph G is added to an arbitrary tree t(a,r), then the resulting graph is a 1-tree of the original graph. This observation suggests that the complete set of 2-tree t(a,r) can be obtained from the set of 1-tree of the graph by first sorting the subset of 1-trees which contain the edge between nodes a and r, then next deleting this edge from every member of the subset. If in the original graph nodes a and r are not directly connected, then the indicated procedure can be applied on the augmented graph G_a derived from G by adding a fictituous edge between nodes a and r. In terms of the edge T-matrix, the Type-I 2-tree summation can be obtained as follows.

Theorem 5.4 Given a graph G, let a and r be two specified nodes of G. The Type-I 2-tree summation T(a,r) of G can be determined through application of the following steps:

Step 1. Obtain from the original graph its edge T-matrix T. The augmented graph G_a is used instead of G if nodes a and r are disconnected. Using Definition 3.1-a for directed graphs, after deleting all outgoing edges from a and r; and using Definition 3.1-b for nonoriented graph.

Step 2. Generate the set of T-subsequent edge T-matrices from $\lfloor \underline{T}$ by using the T-transform. Applying Rule 5.1 if necessary. From each member of the set of T-subsequent edge T-matrices, $\lfloor \underline{T}_m$ compute the partial Type-I 2-tree summation as given below:

$$\left| T(a,r)_m \right|_p = \begin{cases} \prod_{i=1 \neq f}^{n-1} f_i^m, & \text{if } t_{fq} \in s_f^m \\ 0, & \text{if } t_{fq} \notin s_f^m \end{cases} \qquad (5.5)$$

where t_{fq} is the entry associated with the edge between nodes a and r in the edge T-matrix $\lfloor \underline{T}_m$ under consideration. For an oriented graph, replace s_i^m by S_i^m and applying the *multiplication in Equation 5.5.

Step 3. Calculate the sum of all of the partial Type-I 2-tree summations obtained in Step 2.

Proof: If the entry t_{fq} associated with the edge between nodes a and r is not present in $\lfloor \underline{T}_m$, then it is clear that none of the 1-trees obtained from the partial 1-tree summation evaluated from $\lfloor \underline{T}_m$ does contain this edge, then it is clear that none of the tree t(a,r) can be obtained from $\lfloor \underline{T}_m$. Suppose now that the edge between a and r is associated with the entry t_{fq} of $\lfloor \underline{T}_m$, then the tree products of all of the trees which contain that edge obtained from $\lfloor \underline{T}_m$ can be determined by the product

$$t_{fq} \prod_{i=1,\neq f}^{n-1} s_f^m$$

Deleting the edge between nodes a and r from these 1-tree is equivalent with letting $t_{fq} = 1$ in this product.

Therefore, $\left|T(a,r)_m\right|_p$ is the partial Type-I 2-tree summation of $\underline{|T_m}$. The Type-I 2-tree is clearly determined by summing all $\left|T(a,r)_m\right|_p$ by the light of Theorem 5.3.

If the graph under consideration is a directed graph, the deletion of all out-going edges from nodes a and r is necessary, since these nodes are considered as reference nodes in each subgraph of t(a,r), and replacing s_f^m by S_f^m and applying the *multiplication will systematically eliminate all non conform directed Type-I 2-trees from the partial Type-I 2-tree $\left|T(a,r)_m\right|_p$.

Remark 5.4 It is noted that the Type-I 2-tree summation can be obtained at the same time with the 1-tree summation by applying the algorithm given in Section 5.3.1 and Theorem 5.4. This can be seen from the fact that, supposing nodes a and r are directly connected in the original graph, the 1-tree summation and the Type-I 2-tree summation are both determined partially from every member of the set of T-subsequent edge T-matrices derived from the original T-matrix $\underline{|T_m}$. Then if in Step 2 of the algorithm of Section 5.3.1, the partial 1-tree summation $\left|T_m\right|_p$ and the partial Type-I 2-tree summation $\left|T(a,r)_m\right|_p$ are evaluated, then in Step 3, these partial tree summations are summed separately the 1-tree summation and the Type-I 2-tree summation will be obtained at the same time. If now Theorem 5.4 is applied to the augmented graph G_a, and in Step 2 the partial 1-tree summation of the original graph can be obtained as follows:

$$\left|T_m\right|_p = \begin{cases} s_f^{m'} \prod_{i=1,\neq f}^{n-1} s_i^m, & \text{if } t_{fq} \in s_f^m \\ \prod_{i=1}^{n-1} s_i^m, & \text{if } t_{fq} \notin T_m \end{cases} \qquad (5.6)$$

where $s_f^{m'}$ is the sum of all entries in row f of $\left|T_m\right|$ with $t_{fq} = 0$ where t_{fq} is the entry associated with the edge added to the original graph between nodes a and r. Equation 5.6 can be proved by noting that letting $t_{fq} = 0$ is equivalent with deleting the added edge from the augmented graph. Next, in Step 3 the 1-tree summation and the Type-I 2-tree summation will be obtained at the same time by summing separately all of the partial 1-tree summations and the partial Type-I 2-tree summation obtained in Step 2.

Example 5.2 Evaluate the Type-I 2-tree of the graph shown in Figure 5.1-a, with node 1 in one subgraph and node 4 in the other.

Step 1. Obtain the edge T-matrix from the graph (see Example 5.1).

Step 2. Generate the set of T-subsequent edge T-matrix (see example 5.1). Compute the partial Type-I 2-tree from each member of this set:

From $\left|T_1\right|$: $\left|T(1,4)_1\right|_p = (2,b)+(1,a)\ (3,g)+(3,g)+(2,d)$

$= bg + bc$

From $\left|T_{12}\right|$: none since edge f is not present in T_{12}

From $\left|T_{12^2}\right|$: none by the same reason

From $\left|\underline{T_{13}}: \left|T(1,4)_{13}\right|_p\right.$ = (3,g)+(3,c) (2,d)

= gd + dc

From $\left|\underline{T_{1(2)^2 3}}\right.$: none by the same reason

Step 3. The Type-I 2-tree summation of the given graph is

$$T(1,4) = gd + dc + bg + bc.$$

The obtained Type-I 2-trees are shown in Figure 5.2.

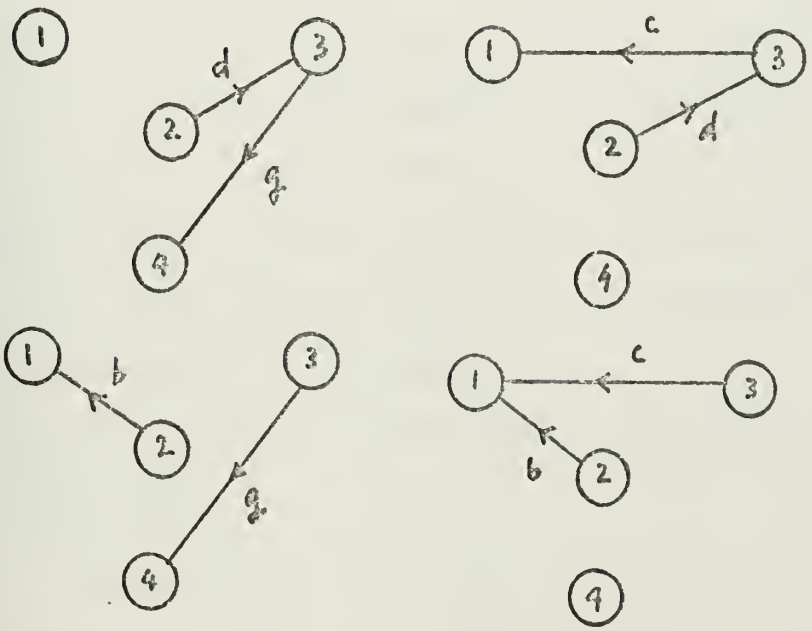


Figure 5.2. Illustration of Theorem 5.4.

E. EVALUATION OF TYPE-II 2-TREE SUMMATION

Consider a node triplet a,b and r of a given graph, G. In this section a technique for computing the Type-II 2-tree summation, T(a,b;r) of the graph G, will be derived. In this order, consider the following formula, shown by Farber and Malik [12].

$$T(a,b;r) = T(a,r) \bigcap T(b,r) \tag{5.7}$$

Applying Theorem 5.4 and Eq. 5.7, it results that $T(a,b;r)$ can be obtained by computing partially $|T(a,b;r)_m|_p$, from $|T(a,r)_m|_p$ by eliminating all of the trees obtained in the later partial Type-I 2-tree summation, of which nodes b and r are connected.

1. Evaluation of Partial Type-II 2-tree Summation

Suppose that node r of G is associated with row r, and node a and b with column 1 and b of the original edge T-matrix of G. Let t_{f1} and t_{xy} be the entries of an arbitrary T-subsequent edge T-matrix derived from $|\underline{T}$, which correspond to the edges connecting nodes a and r, and b and r, respectively. Depending on the location of t_{f1} and t_{xy} in the edge T-matrix $|\underline{T}_m$ under consideration, there are three cases to be investigated.

Case 1. If t_{f1} is eliminated by the process of generating $|\underline{T}_m$, then

$$|T(a,b;r)_m|_p = 0 \tag{5.8.a}$$

Eq. 5.8.a results from the fact that $|T(a,b;r)_m|_p$ is evaluated from $|T(a,r)_m|_p$, and the later is zero.

Case 2. If t_{f1} and t_{xy} are both in the same row f of $|\underline{T}_m$. Then

$$|T(a,b;r)_m|_p = \prod_{i=1,\neq f}^{n-1} s_i^m \tag{5.8.b}$$

Eq. 5.8.b can be obtained by noting that its right hand side member can be considered as expressing the Type-I

2-tree summation $|T(a,r)_m|_p$ and $|T(b,r)_m|_p$ obtained from $|\underline{T_m}$. This implies that all of 2-trees obtained in $|T(a,r)_m|_p$ have node r in one subtree and nodes a and b in the other. Hence Eq. 5.8.b.

Case 3. If t_{fl} and t_{xy} are not in the same row of $|\underline{T_m}$. Let s_i^{m*} be the sum of entries in row i, and in the first f columns. Then, with $f+1 \leq k \leq x$,

$$|T(a,b;r)_m|_p = s_x^{m*} \prod_{i=1,\neq f,x}^{n-1} s_i^m + \sum_{k=f+1}^{x} t_{xk} s_{k-1}^{m*} \prod_{i=1,\neq f,k-1,x}^{n-1} s_i^m \qquad (5.8.c)$$

To prove Eq. 5.8.c, expand the expression of $|T(a,r)_m|_p$ with respect to the sum s_x^m.

$$|T(a,r)_m|_p = s_x^{m*} \prod_{i=1,\neq f,x}^{n-1} s_i^m + \sum_{k=f+1}^{x} t_{xk} s_{k-1}^{m*} \prod_{i=1,\neq f,k-1,x}^{n-1} s_i^m + \sum_{k=f+1}^{x} t_{xk} s_{k-1}^{m'} \prod_{i=1,\neq f,k-1,x}^{n-1} s_i^m \qquad (5.9)$$

where

$$s_i^{m'} = s_i^m - s_i^{m*} \qquad (5.9.a)$$

Comparing Eq. 5.9 with Eq. 5.8.c, shows that only terms in the third summation of Eq. 5.9 are eliminated, in other words this summation is the summation of tree products of the Type-I 2-trees obtained from $|T(a,r)_m|_p$ of which nodes b and r are both in the same subtree, and the first and

second summation are the summations of tree products of Type-I 2-tree obtained from $\left|T(a,r)_m\right|_p$ of which nodes b and r are in different subtrees.

To prove the above statements, the following lemma is to be considered first.

Lemma 5.1 Consider a graph G and two specified nodes, r and b of G. Let $\left|\underline{T}_m\right.$ be an arbitrary member of the set of T-subsequent edge T-matrices associated with G. Suppose that in $\left|\underline{T}_m\right.$ node r and b are associated with row f and column f+1 and row x and column x+1, respectively. Furthermore, suppose that all of the entries in row f are zero, and $f < x$. Then a path connecting nodes r and b can be represented by

$$P(i_1) = t_{i_1,f+1}\, t_{i_2,i_1+1}\, t_{i_3,i_2+1} \cdots t_{i_k,i_{k-1}+1}\, t_{x,i_k+1} \quad (5.10)$$

with

$$f+1 \leq i_1,\ i_1+1 \leq i_2,\ldots,i_j+1 \leq i_{j+1},\ldots,i_k+1 \leq x$$

if the path is formed by taking one entry in each row of $\left|\underline{T}_m\right.$ at a time.

Proof: $P(i_1)$ is clearly the path product of a path which connects nodes r and b, since the nodes i_j-1, contained in this path, and associated with row i_j, with $1 \leq j \leq k$, are all of degree two and nodes r and b, associated with column f+1 and row x, respectively, are both of degree one. The inequalities between values of i_j must be satisfied such that the condition that edges contained in the paths correspond to entries taken one at a time in

a row of $\lfloor T_m$. These inequalities shows that entries forming $P(i_1)$ should be taken one from row f down to row x in such a way that if the preceding entry has been taken from a row, say i_j, then the next entry must be selected in column i_j+1 and in any row below row i_j, and above row x. If this condition is not satisfied, then the path does not exist. This can be seen by supposing that, if $i_1 \leq f+1$, and the sequence of rows, i_j, from which entries of $P(i_1)$ are selected is decreasingly ordered, then the second entry should be taken from row f. This is impossible, since it has been supposed that entries in this row are zero. Furthermore, the sequence of row i_j from which entries in $P(i_1)$ are selected must be uniformly and increasingly ordered as indicated. This can be proven as follows. Suppose that this sequence is uniformly and increasingly ordered from i_1 successively to i_j. The last entry under consideration is associated with an edge connected to node associated to row i_j. The next entry should be an entry which represents an edge connecting to this node. All of entries which correspond to such edges are located in row i_j and in column i_j+1. Since one entry in row i_j has been selected, then the entry which is being selected must be located in column i_j+1, and since the first row in column i_j+1 is row i_j+1, therefore the sequence of rows from which entries of $P(i_1)$ must be uniformly and increasingly ordered.

Lemma 5.2 Consider a graph G and two specified nodes r and b of G. Let $\lfloor T_m$ be an arbitrary member of the set of

T-subsequent edge T-matrices associated with G. Suppose that in $\lfloor T_m$ nodes r and b correspond to rows f and row x respectively. Furthermore entries in row f are all zero, and $f < x$. Then paths $P(i_1)$ do not exist if:

(a) in row x, the unique non zero entry is t_{xk^*}, and if $k^* < f+1$.

(b) in row x the unique non zero entry is t_{xk}, and all entries in row k-1 are zero.

Proof: If $k^* < f+1$, then the sequence of rows i_j from which entries of the paths are selected can not be obtained. Hence Lemma 5.2.a is proved. Lemma 5.2.b results from Lemma 5.1 directly by noting that, corresponding to every value of i_1, the entry preceding the last entry in $P(i_1)$ should be an entry in row k-1. Then if all entries in row k-1 are zero, then $P(i_1)$ can not exist.

Eq. 5.8.c can now be proved by applying Lemma 5.2. In this order, it is observed that all entries t_{xk} of $s_x^{m^*}$ are in column k, with $1 \leq k \leq f$. Then all 2-trees obtained by expanding the first product of Eq. 5.9 are Type-II 2-trees defined with respect to the node triplet a, b and r, since in these 2-trees nodes b and r are not in the same subtree. This can be verified by noting that with respect to every t_{xk} of $s_x^{m^*}$ all of the conditions of Lemma 5.2.a are verified. Similarly, with respect to every t_{xk} in the second term of Eq. 5.9, the conditions of Lemma 5.2.b are also satisfied. This is seen by noting that in the sum $s_{k-1}^{m^*}$ only entries $t_{k-1,z}$, with $1 \leq z \leq f$ are

considered. Then this summation represents a summation of tree products of 2-trees of $|T(a,r)_m|_p$ which can be considered as Type-II 2-trees defined with respect to nodes a, b and r. Finally, expanding the third term in Eq. 5.9, it is verified that each term in the resulting sum contain a product which can be considered as a path $P(i_1)$. Therefore all of these terms are tree products of those tree of $|T(a,r)_m|_p$ of which node b and r are in the same subtree. These tree should be eliminated in the computation of $|T(a,b;r)_m|_p$. Hence Eq. 5.8 is prove.

Example 5.3 Evaluate the partial Type-II 2-tree summation, T(1,4;9) in the following edge T-matrix

$$|\underline{T(2^3 3^2)} = \begin{array}{c|l} 5 & d \\ 8 & g\ c \\ 9 & h\ D\ J \\ 2 & .\ k\ n\ o \\ 3 & .\ q\ t\ u\ i \\ 4 & .\ v\ y\ z\ j\ p \\ 6 & .\ .\ F\ G\ l\ r\ w \\ 7 & .\ .\ H\ I\ m\ s\ x\ E \\ \hline & 1\ 5\ 8\ 9\ 3\ 4\ 6 \end{array}$$

The dots are used in this edge T-matrix instead of zero for better legibility. The edges connecting nodes 1 to 9 and 4 to 9 are associated with entries h and z, respectively. They are located in different rows, then Eq. 5.8-c is now applied. From the 6^{th} row, which contains z, and associated with node 4, it is found:

$$s_6^{m*} = v+y,\ t_{65} = j,\ t_{66} = p.$$

Corresponding to t_{65} and t_{66}, are

$$m_4^* = k+n \quad \text{and} \quad m_5^* = q+t$$

Then:

$$\left|T(1,4;9)_{2^3 3^2}\right|_p = (v+y)d(g+c)(k+n+0)(q+t+u+i)(F+G+1+r+w)(H+I+m+s+x+E)$$
$$+ j(k+n)d(g+c)(q+t+u+i)(F+G+1+r+w)(H+I+m+s+x+E)$$
$$+ p(q+t)d(g+c)(k+n+o)(F+G+1+r+w)(H+I+m+s+x+E)$$

Remark 5.5 In the forgoing discussion, it has been supposed that both of the entries corresponding to edges connecting nodes r to nodes a and b are included in $\lfloor T_m$. It is also assumed that the row f associated with the entry associated with edge between r and a is located above the one x, containing the entry associated with the edge between r and b. It will be shown that no generality is lost in this restriction. Although row f could be located below row x, but it is easy to verify that when that situation does occur, then (a) the entry corresponding to the edge between r and a will be eliminated by the process of generating the new edge T-matrix, and as shown in Case 1, the partial Type-I 2-tree summation computed from the newly generated edge T-matrix is zero and (b) t x y and t + 1 are in the same row, that is Case 2. Therefore it is not necessary to investigate the case where row f is situated below row x. Next, it is true that the entry associated with the edge between nodes r and b could be eliminated in the generation of subsequent

T-matrices. Then it is desirable to design a method for keeping track of row x in each newly generated edge T-matrix. In general this can be performed by using the node vector defined below.

Definition 5.3 The node vector, N_m, is a row vector, of which the entry, n_j^m, is defined equal to the node associated with column j, and n_n^m the node associated with row n-1 of the edge T-matrix $\lfloor T_m$ corresponding to N_m.

Property 5.1 Given an edge T-matrix $\lfloor T_m$ and its corresponding node vector N_m. Let $\lfloor T_{mj}$ be the edge T-matrix derived from $\lfloor T_m$ by applying the T-transformation with respect to column j of $\lfloor T_m$. Then, the node vector N_{mj} of $\lfloor T_{mj}$ can be obtained as:

$$n_i^{mj} = \begin{cases} n_i^m & \text{for } 1 \leq i \leq j-1 \\ n_{i+1}^m & \text{for } j \leq i \leq n-1 \\ n_j^m & \text{for } i = n \end{cases}$$

Property 5.1 results obviously from Remark 5.1.

2. Algorithm for Evaluating Type-II 2-tree Summation

The forgoing discussion can now be summarized by the following algorithm.

Theorem 5.5 Given a graph G and let a, b, r be three specified node of G. The Type-II 2-tree summation T(a,b;r) of G can be determined through application of the following steps:

Step 1. Obtain from the original graph G, its edge T-matrix $\lfloor \underline{T}$. The augmented graph G_a is used instead of G if nodes a and r are disconnected. Using Definition 3.1-a for directed graphs, after deleting all outgoing edges from nodes a and r. Definition 3.1-b is used for non-oriented graph. Form the node vector corresponding to $\lfloor \underline{T}$.

Step 2. Generate the set of T-subsequent edge T-matrices from $\lfloor \underline{T}$, together with their node vector, using the T-transformation and Property 5.1. From each member $\lfloor \underline{T}_m$ of this set, compute the partial Type-II 2-tree summation $T(a,b;r)_{m\ p}$, by applying Eqs. 5.8.a, b, and c depending on the relative location of row f and row x in $\lfloor \underline{T}_m$. For oriented graphs, replace s_i^m by S_i^m and applying the *multiplication in Eq. 5.8.

Step 3. Calculate the sum of all of the partial Type-II 2-tree summations obtained in Step 2.

Example 5.4 Evaluate the Type-II 2-tree summation T(1,3;5) of the complete fine-node graph shown in Figure 5.3.

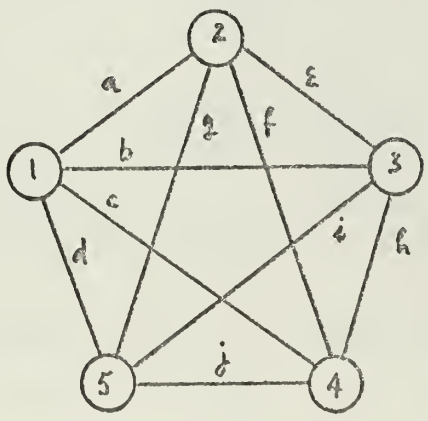


Figure 5.3.a. Illustration of Theorem 5.5.

Step 1 The edge T-matrix and the node vector associated with the graph are

$$\lfloor\underline{T} = \begin{vmatrix} a & & & \\ b & e & & \\ c & f & h & \\ d & g & i & j \end{vmatrix}$$

$$N = 1\ 2\ 3\ 4\ 5$$

Step 2. Generation of the set of T-subsequent edge T-matrices derived from $\lfloor\underline{T}$ and their node vectors. For saving space, only edge T-matrices from which non zero partial Type-II 2-tree summation can be obtained will be listed below.

$$\lfloor\underline{T_2} = \begin{vmatrix} b & & & \\ c & h & & \\ d & i & j & \\ . & e & f & g \end{vmatrix} \qquad \lfloor\underline{T_2{}^2} = \begin{vmatrix} c & & & \\ d & j & & \\ . & f & g & \\ . & h & i & e \end{vmatrix} \qquad \lfloor\underline{T_3} = \begin{vmatrix} a & & & \\ c & f & & \\ d & g & j & \\ . & . & h & i \end{vmatrix}$$

$$N_2 = 1\ 3\ 4\ 5\ 2 \qquad N_2{}^2 = 1\ 4\ 5\ 2\ 3 \qquad N_3 = 1\ 2\ 4\ 5\ 3$$

$$\lfloor\underline{T_{23}} = \begin{vmatrix} b & & & \\ d & i & & \\ . & e & g & \\ . & . & j & f \end{vmatrix} \qquad \lfloor\underline{T_4} = \begin{vmatrix} a & & & \\ b & e & & \\ d & g & i & \\ . & . & . & j \end{vmatrix}$$

$$N_{23} = 1\ 3\ 5\ 2\ 4 \qquad N_4 = 1\ 2\ 3\ 5\ 4$$

$$\lfloor\underline{T_{234}} = \begin{vmatrix} b & & & \\ d & i & & \\ . & . & j & \\ . & . & . & f \end{vmatrix} \qquad \lfloor\underline{T_2{}^2{}_4} = \begin{vmatrix} c & & & \\ d & j & & \\ . & h & i & \\ . & . & . & e \end{vmatrix}$$

$$N_{234} = 1\ 3\ 5\ 4\ 2 \qquad N_2{}^2{}_4 = 1\ 4\ 5\ 3\ 2$$

Listing of patial Type-II 2-tree summations.

Eq. 5.8.b is applied to $\lfloor\underline{T_1}$, $\lfloor\underline{T_2}$, $\lfloor\underline{T_{23}}$, $\lfloor\underline{T_4}$.

From $\underline{|T_1}$: $|T(1,3;5)_1|_p = a(b+e)(c+f+h)$

From $\underline{|T_2}$: $|T(1,3;5)_2|_p = b(c+h)(e+f+g)$

From $\underline{|T_4}$: $|T(1,3;5)_4|_p = a(b+e)j$

From $\underline{|T_{23}}$: $|T(1,3;5)_{23}|_p = b(e+g)(j+f)$

Eq. 5.8.c is applied to $\underline{|T_2{}^2}$, $\underline{|T_3}$, $\underline{|T_{234}}$, $\underline{|T_2{}^2{}_4}$

From $\underline{|T_2{}^2}$: $|T(1,3;5)_{2^2}|_p = hc(f+g) + efc$

From $\underline{|T_3}$: $|T(1,3;5)_3|_p = ha(c+f)$

From $\underline{|T_{234}}$: $|T(1,3;5)_{234}|_p = bjf$

From $\underline{|T_{2^24}}$: $|T(1,3;5)_{2^24}|_p = ehc$

Summing all of the obtained partial Type-II 2-tree summation gives the Type-II 2-tree summation, T(1,3;5), which is composed of 25 2-trees shown in Figure 5.3.b.

F. EVALUATION OF TYPE-III AND TYPE-IV 2-TREE SUMMATIONS

Consider four nodes, a, b, c, and r of a given non-oriented graph G. In this section, technique will be derived for computing simultaneously the type-III and type-IV 2-tree summations, denoted by T(a,b;r,c) and T(a,b,c;r), respectively. In this order, the following formulas [20] are considered.

$$T(a,b;r) = T(a,b;r,c) + T(a,b,c;r) \qquad (5.11.a)$$

$$T(a,c;r) = T(a,c;r,b) + T(a,b,c;r) \qquad (5.11.b)$$

It results from Eqs. 5.11

$$T(a,b,c;r) = T(a,b;r) \bigcap T(a,c;r) \qquad (5.12.a)$$

$$T(a,b;r,c) = T(a,b;r) - T(a,b,c;r) \qquad (5.12.b)$$

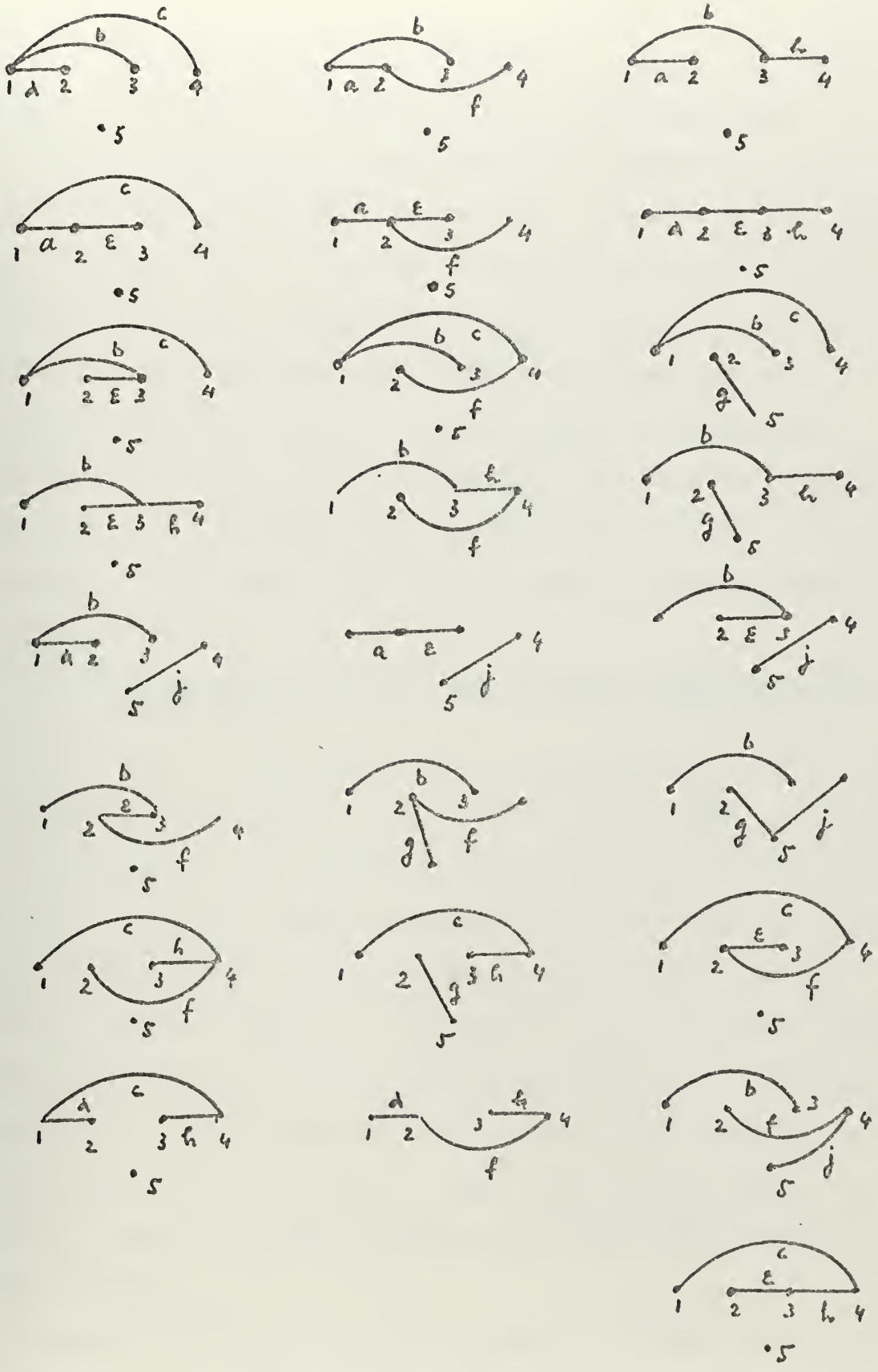


Figure 5.3.b. Set of Type-II 2-trees Defined with Respect to Nodes 1, 3 and 5 from a Complete Pentagon.

$$T(a,c;r,b) = T(a,c;r) - T(a,b,c;r) \qquad (5.12.c)$$

From Eq. 5.12, and Theorem 5.5 it follows that, since both of T(a,b;r) and T(a,c;r) are partially determined from a set of T-subsequent edge T-matrices of G which contain the entry corresponds to the edge between node a and r, then their intersection, namely T(a,b,c;r) can also be evaluated partially from the same edge T-matrices together with T(a,b;r,c) and T(a,c;r,b). The problem of evaluating Type-III and Type-IV 2-tree summation can then be reduced to the problem of determining the intersection of two Type-II 2-tree partial summations obtained from the same T-subsequent edge T-matrix.

1. Evaluation of Partial Type-III and Type-IV 2-tree Summations

Suppose that node r of G is associated with row n-1, and node a with column 1, and b and c with columns 2 and 3 respectively, in the original edge T-matrix corresponding to G. Let t_{f1} be the entry associated with the edge connecting nodes a and r, (an additive edge, if necessary), in an arbitrary edge T-matrix T_m of the set of T-subsequent edge T-matrices derived from G, and from which t_{f1} has not been eliminated. Then there are three cases to be considered, depending on the locations of t_{f1}, t_{uv}, and t_{xy} where t_{uv} and t_{xy} represent edges between node r and nodes b and c respectively.

Case 1. If t_{f1}, t_{uv}, and t_{xy} are all in the same row f of $\lfloor T_m$, then from Eq. 5.8.b

$$\left|T(a,b;r)_m\right|_p = \left|T(a,c;r)_m\right|_p$$

and from Eq. 5.12

$$\left|T(a,b,c;r)_m\right|_p = \prod_{i=1,\neq f}^{n-1} s_i^m \qquad (5.13.a)$$

and

$$\left|T(a,b;r,c)_m\right|_p = \left|T(a,c;r,b)_m\right|_p = 0 \qquad (5.13.b)$$

Case 2. If in $\left|T_m\right|$, t_{uv} and t_{xy} are not in the same row, and located in rows u and x, respectively. Then the partial Type-III and Type-IV 2-tree summations can be obtained as follows. From Eq. 5.8.c:

$$\left|T(a,b;r)_m\right|_p = s_u^{m*} \prod_{i=1,\neq f,u}^{n-1} s_i^m + \sum_{h=f+1}^{u} t_{uh} s_i^{m*} \prod_{i=1,\neq f,u,h-1}^{n-1} s_i^m \qquad (5.14.a)$$

$$\left|T(a,c;r)_m\right|_p = s_x^{m*} \prod_{i=1,\neq f,x}^{n-1} s_i^m + \sum_{k=f+1}^{x} t_{xk} s_{k-1}^{m*} \prod_{i=1,\neq f,x,k-1}^{n-1} s_i^m \qquad (5.14.5)$$

To evaluate $\left|T(a,b,c;r)_m\right|_p$ rewrite Eq. 5.14.a as follows:

$$\left|T(2,b;r)_m\right|_p = s_u^{m*} s_x^{m*} \prod_{i=1,\neq f,u,x}^{n-1} s_i^m +$$

$$\sum_{k=f+1}^{x} t_{xk} s_u^{m*} s_{k-1}^{m'} \prod_{i=1,\neq f,u,x,k-1}^{n-1} s_i^m + \sum_{k=f+1}^{x} t_{xk} s_u^{m*} s_{k-1}^{m*} \prod_{i=1,\neq f,u,x,k-1}^{n-1} s_i^m$$

$$\sum_{h=f+1}^{u} t_{uh} s_x^{m*} s_{h-1}^{m*} \prod_{i=1,\neq f,u,x,h-1}^{n=1} s_i^m + \sum_{h=f+1}^{u} t_{uh} s_x^{m'} s_{h-1}^{m*} \prod_{i=1,\neq f,u,x,h-1}^{n-1} s_i^m \qquad (5.14.c)$$

In Eq. 5.14.c, the first three terms are obtained by taking the sum s_x^m out of its finite product, next applying Eq. 5.9.a to write:

$$s_x^m = s_x^{m'} + s_x^{m*} \qquad (5.14.d)$$

then, corresponding to each term t_{xu} in $s_x^{m'}$, take the sum s_{k-1}^m out of the finite product and separate it into two sums as in Eq. 5.14.d; the last two terms are obtained by taking the sum s_x^m out of the finite product, then using Eq. 5.14.d to write the second term of Eq. 5.14.a into two terms.

Since Eq. 5.14.b can be derived from Eq. 5.14.a by interchanging the subsequents u, h and x, k, then Eq. 5.14.b can be also rearranged into an equation similar to Eq. 5.14.c, by performing the same operation of subscripts interchanging. Therefore

$$\left|T(a,b,c;r)_m\right|_p = s_u^{m*} s_x^{m*} \prod_{i=1,\neq f,u,x}^{n-1} s_i^m +$$

$$\sum_{k=f+1}^{x} t_{xk} s_u^{m*} s_{k-1}^{m*} \prod_{i=1,\neq f,u,x,k-1}^{n-1} s_i^m +$$

$$\sum_{h=f+1}^{u} t_{uh} s_x^{m*} s_{h-1}^{m*} \prod_{i=1,\neq f,u,x,h-1}^{n-1} s_i^m \qquad (5.15.a)$$

Eq. 5.15.a is obtained from Eq. 5.14.c by selecting terms in which the subscripts u and x are symmetric and terms which can be from each other by interchanging subscripts x, k-1 and u, h-1, respectively. From Eqs. 5.14.c and 5.15.a, it results

$$\left|T(a,b;r,c)_m\right|_p = \sum_{h=f+1}^{u} s_x^{m'} t_{uh} s_{h-1}^{m*} \prod_{i=1,\neq f,u,x,h-1}^{n-1} s_i^m +$$

$$\sum_{k=f+1}^{x} s_u^{m*} s_{k-1}^{m'} t_{xk} \prod_{i=1,\neq f,u,x,k-1}^{n-1} s_i^m \qquad (5.15.b)$$

Eq. 5.15.b is obtained by subtracting Eq. 5.15.a from Eq. 5.14.c. Furthermore by the symmetry observed above of the subscripts u, h and x, k, it is clear that

$$\left|T(a,c;r,b)_m\right|_p = \sum_{k=f+1}^{x} s_u^{m'} t_{xk} s_{k-1}^{m*} \prod_{i-1,\neq f,u,x,k-1}^{n-1} s_i^m +$$

$$\sum_{h=f+1}^{u} s_x^{m*} s_{h-1}^{m'} t_{uh} \prod_{i=1,\neq f,u,x,h-1}^{n-1} s_i^m \qquad (5.15.c)$$

Case 3. If t_{fl} and t_{uv} are in the same row f of $\left|T_m\right|$ which does not contain t_{xy}. Then, from Eq. 5.8.b:

$$\left|T(a,b;r)_m\right|_p = \prod_{i=1,\neq f}^{n-1} s_i^m \qquad (5.16.a)$$

and $\left|T(a,c;r)_m\right|_p$ is determined by Eq. 5.14.b. Noting that all terms of Eq. 5.14.b are included in Eq. 5.16.a, then

$$\left|T(a,c;r,b)_m\right|_p = 0 \qquad (5.16.b)$$

and

$$\left|T(a,b,c;r)_m\right|_p = s_x^{m*} \prod_{i=1,\neq f,x}^{n-1} s_i^m + \sum_{k=f+1}^{x} t_{xk} s_{k-1}^{m*} \prod_{i=1,\neq f,x,k-1}^{n-1} s_i^m$$

(5.16.c)

Furthermore, $\left|T(a,b;r,c)_m\right|_p$ can be obtained by subtracting Eq. 5.16.c from Eq. 5.16.a. Therefore:

$$\left|T(a,b;r,c)_m\right|_p = \sum_{k=f+1}^{x} t_{xk} s_{k-1}^{m'} \prod_{i=1,\neq f,k-1}^{n-1} s_i^m \qquad (5.16.d)$$

Similarly, if t_{f1} and t_{xy} are in the same row f of $\underline{T_m}$ which does not contain t_{uv}, then:

$$\left|T(a,b;r,c)_m\right|_p = 0 \qquad (5.17.a)$$

$$\left|T(a,b,c;r)_m\right|_p = s_u^{m*} \prod_{i=1,\neq f,x}^{n-1} s_i^m + \sum_{h=f+1}^{u} \sum t_{uh} s_{h-1}^{m*} \prod_{i=1,\neq f,x,h-1}^{n-1} s_i^m \qquad (5.17.b)$$

and

$$\left|T(a,c;r,b)_m\right|_p = \sum_{h=f+1}^{u} t_{uh} s_{h-1}^{m'} \prod_{i=1,\neq f,u,h-1}^{n-1} s_i^m \qquad (5.17.c)$$

Remark 5.6 In the above discussion it is supposed that t_{uv} and t_{xy} are both contained in $\underline{T_m}$. In the process of generating $\underline{T_m}$, one or both of these entries can be eliminated. The node vector can then be used to locate rows x and u.

Remark 5.7 The finite products involved in the above discussion are defined equal to one, if no s_i^m remains after excluding all of the indicated s_i^m.

2. Algorithm for Evaluating Type-III and Type-IV 2-tree Summations

The forgoing discussion can now be summarized in the next theorem for the generation of Type-III and Type-IV 2-tree summations.

Theorem 5.6 Given a graph G, let a,b,c and r be four specified nodes of G. The Type-III 2-tree summations T(a,b; r,c) and T(a,c;r,b) and the Type-IV 2-tree summation T(a,b,c;r) can be determined through the applications of the following steps.

Step 1. Obtain from the original graph G its edge T-matrix $\lfloor \underline{T}$. The augmented graph G_a is used instead of G if nodes a and r are disconnected. Using Definition 3.1.a for directed graphs, after deleting all outgoing edges from nodes serving as reference nodes in each subtree. From the node vector corresponding to $\lfloor \underline{T}$.

Step 2. Generate the set of T-subsequent edge T-matrices and their node vector, using the T-transformation and Property 5.1. From each member $\lfloor \underline{T}_m$ of this set, compute the partial Type-III and Type-IV 2-tree summations $\left\lfloor T(a,b;r,c)_m \right\rfloor_p$, $\left\lfloor T(a,c;r,b)_m \right\rfloor_p$ and $\left\lfloor T(a,b,c;r)_m \right\rfloor_p$ by applying Eq. 5.13, Eq. 5.15 or Eq. 5.16 b, c and d, and Eq. 5.17, depending on the relative location of rows f, u, and x in $\lfloor \underline{T}_m$.

For directed graphs, replace s_i^m by S_i^m and applying the *multiplication.

Step 3. Calculate the sum of all of the partial Type-III and Type-IV 2-tree summation obtain in Step 2.

Example 5.5 Evaluate the Type-III 2-tree summations T(1,2;5,3) and T(1,3;5,2) and the Type-IV 2-tree summation T(1,2,3;5) of the complete five-node graph of Figure 5.3.

Step 1. (See example 5.4).

Step 2. Generation of the set of T-subsequent edge T-matrices derived from the original edge T-matrix and their node vectors, (see Ex. 5.4). Type-III 2-tree partial summation $|T(1,2;5,3)_m|_p$ and $|T(1,3;5,2)_m|_p$ and Type-IV 2-tree partial summation $|T(1,2,3;5)_m|_p$ computed from every member of the set of T-subsequent edge T-matrices are listed and tabulated in Table 5.1 below.

Step 3. Total Type-III 2-tree summations and Type-IV 2-tree summation are given in Table 5.2.

TABLE 5.1

LIST OF $|T(1,2;5,3)_m|_p$, $|T(1,3;5,2)_m|_p$ AND $|T(1,2,3;5)_m|_p$

$\lfloor T_m$	$\lvert T(1,2;5,3)_m \rvert_p$	$\lvert T(1,3;5,2)_m \rvert_p$	$\lvert T(1,2,3;5)_m \rvert_p$
T_1	0	0	a(b+e)(c+f+h)
T_2	0	b(c+h)g	b(c+h)(e+f)
T_{22}	cfi	hcg	cf(e+h)
T_3	ai(c+f)	0	ha(c+f)
T_{23}	0	bg(j+f)	be(j+f)
T_4	0	0	a(b+e)j
T_{234}	0	bjf	0
T_{224}	0	0	che
T_{32}	ai(j+h)	0	0
T_{324}	ajh	0	0

TABLE 5.2

TOTAL TYPE-III 2-TREE SUMMATIONS AND TOTAL TYPE-IV 2-TREE SUMMATION OBTAINED IN EXAMPLE 5.5

$$|T(1,2;5,3)| = cfi + ai(c+f) + ai(j+h) + ajh$$

$$|T(1,3;5,2)| = b(c+h)g + hcg + bg(j+f) + bjf$$

$$|T(1,2,3;5)| = a(b+e)(c+f+h) + b(c+h)(e+f) + cf(e+h) + ha(c+f) + be(j+f) + a(b+e)j + che$$

Type-III 2-trees of $|T(1,2;5,3)|$ and $|T(1,3;5,2)|$ are illustrated in Figure 5.4.a and Figure 5.4.b respectively. For illustration of Type-IV 2-tree of $|T(1,2,3;5)|$, see Figure 5.3.b.

G. EVALUATION OF 1-TREE SUMMATION OF THE GRAPH $G_{14,\,23}$

The problem investigated in this section is the following. Given a graph G, and four of its nodes, 1, 2, 3 and 4. Let $G_{14,23}$ be the graph obtained from G by first identifying the nodes 1 and 4, and then identifying the nodes 2 and 3. Derive an algorithm for computing $|T''|$, the 1-tree summation of $G_{14,23}$ from the set of T-subsequent edge T-matrices of the original graph G. The computation of $|T''|$, is an important problem in topological analysis of two-port network. It will be seen that $|T''|$ is the denominator of all of the elements of the short-circuit admittance matrix of a two-port network, having nodes 1 and 4 as input terminals and nodes 2 and 3 as output terminals as shown in Figure 5.5. It has been shown that [27].

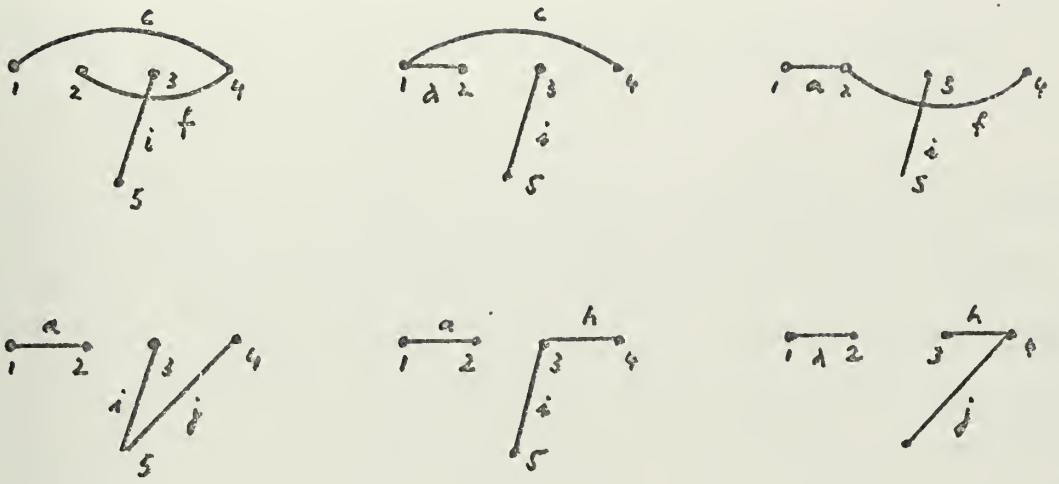


Figure 5.4.a. Set of Type-III 2-trees of T(1,2;5,3) Obtained from a Complete Five-node Graph of Example 5.5.

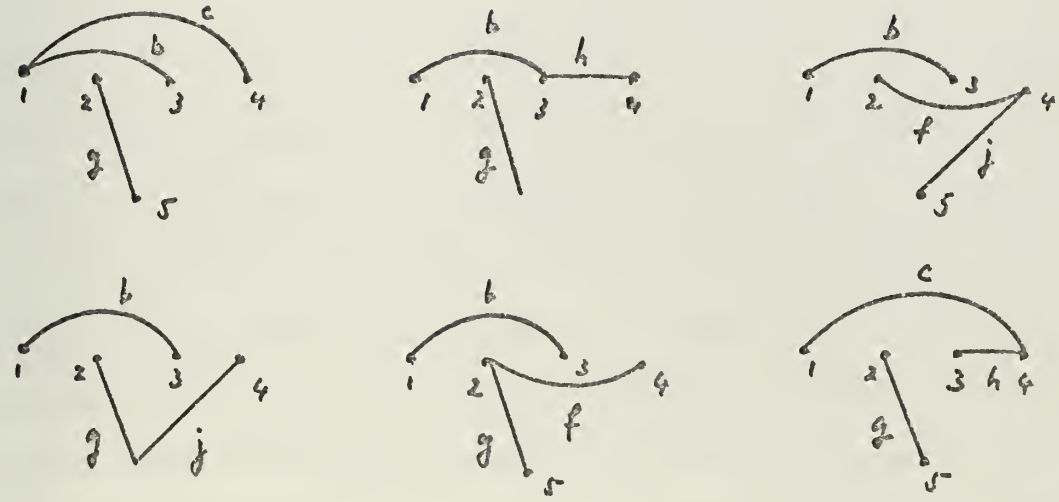


Figure 5.4.b. Set of Type-III 2-trees of T(1,3;5,2) Obtained from a Complete Five-node Graph of Example 5.5.

$$|T''| = |T(1,3;2;4)| + |T(1;2;4,3)| + |T(1,2;3;4)| + |T(1;3;4,2)| \tag{5.18}$$

Suppose that in the original graph G, nodes 1 and 4 are connected, also are nodes 2 and 3. Then Theorem 5.1 applied to the edge e_1 between nodes 1 and 4 gives:

$$|T| = e_1 |T_s| + |T_o| \tag{5.19.a}$$

where $|T_s|$ is the 1-tree summation of the graph derived from G by identifying nodes 1 and 4, $|T_o|$ is that of the graph obtained from G by deleting edge e_i. Applying Theorem 5.1 once again to the edge 2 and 3, with respect to the graph from which T_s is computed. Then

$$|T_s| = e_2 |T_s'| + |T_o'| \tag{5.19.b}$$

where $|T_s'| = |T''|$, since $|T_s'|$ is the 1-tree summation of the graph derived from G by identifying nodes 1 and 4 and then identifying node 2 to 3. Substituting Eq. 5.19.b into Eq. 5.19.a gives

$$|T| = e_1 e_2 |T''| + e_1 |T_o'| + |T_o| \tag{5.19.c}$$

Eq. 5.19.c suggests that $|T''|$ can be obtained by first evaluating a one-tree sub-summation of the graph G, whose trees contain both of the edges e_1 and e_2, then dropping the product $e_1 e_2$ from this 1-tree sub-summation. Since $|T|$ is computed partially from every member of the set of T-subsequent edge T-matrices associated with G. Then an algorithm for computing $|T''|$ can be obtained by deriving a process for evaluating partially $|T''|$ from each member, $|T_m|$ of this set of T-subsequent edge T-matrices. In this order, there are two

cases to be considered, depending on the location of the entries t_{xy} and t_{uv}, which represent the edges e_1 and e_2 respectively, in T_m.

Case 1. If t_{xy} and t_{uv} are on the same row of $\lfloor T_m$, or if either t_{xy} or t_{uv} is not contained in $\lfloor T_m$, or both of them are not contained in $\lfloor T_m$, then:

$$\left|T_m''\right|_p = 0. \qquad (5.20.a)$$

Eq. 5.20.a results directly from the observation that all of the trees obtained from this edge T-matrix, $\lfloor T_m$, do not contain both of the edges e_1 and e_2.

Case 2. If $t_{xy} \varepsilon s_x^m$ and $t_{uv} \varepsilon s_u^m$, then:

$$\left|T_m''\right|_p = \prod_{i=1,\neq,x}^{n-1} s_i^m \qquad (5.20.b)$$

Remark 5.8 It has been shown by Chen that the choice of reference node in $G_{14,23}$ is immaterial. Then for convenience, in the following, nodes 1 and 4 are chosen to be reference nodes. Eq. 5.20, then, can be used for computing T" in the case of a directed graph by replacing in Eq. 5.20.b s_i^m by S_i^m and using the *multiplication, after deleting all outgoing edges from reference nodes and changing all entries $(3,y_k)$ into $(2,y_k)$ in $\lfloor T_m$.

Remark 5.9 In case where fictitious edges have been added to the graph, the partial 1-tree summation of G will be obtained as follows:

1) Delete all additive edges from $\lfloor T_m$ to form $\lfloor T_m'$.

2) Apply Step 2 of the algorithm given in Section 5.3.1 to the resulting subsequent edge T-matrix $|\underline{T_m}'$.

The forgoing discussion can be summarized in the following theorem.

Theorem 5.7 Given a graph G together with four of its nodes, 1, 2, 3 and 4. Let $|T''|$ be the 1-tree summation of the graph $G_{14,23}$ derived from G by identifying nodes 1 and 4, and then nodes 2 and 3. Then $|T''|$ can be obtained through application the following steps.

Step 1. Obtain from the original graph its edge T-matrix $|\underline{T_1}$. The augmented graph is used instead of G if nodes 1 and 4 and/or nodes 2 and 3 are disconnected. Using Definition 3.1.a for directed graph.

Step 2. Generate the set of T-subsequent edge T-matrices derived from $|\underline{T_1}$. From each member $|\underline{T_m}$ of this set, compute the partial summation $|T_m''|_p$ by applying Eqs. 5.20 accordingly to the locations of the entries representing edge e_1 and e_2 in $|\underline{T_m}$.

For directed graphs, delete all of the outgoing edge from nodes 1 and 4, change all $(3,y_k)$ into $(2,y_k)$, apply the *multiplication.

Step 3. Calculate the sum of all $|T_m'|_p$ obtained in step 2.

Example 5.6 Evaluate $|T''|$ of the graph $G_{14,23}$ derived from the five-node complete graph of Figure 5.5.a.

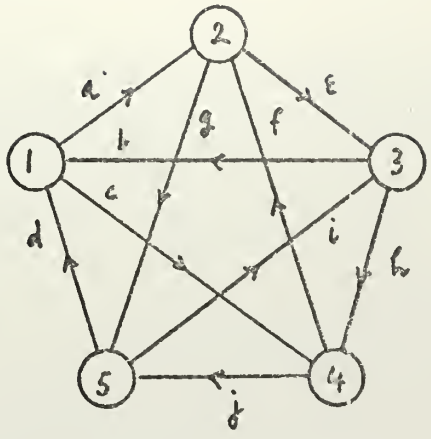


Figure 5.5.a. A Five-node Directed Complete Graph.

Step 1. The edge T-matrix associated with the graph is

$$\underline{|T_1|} = \begin{vmatrix} (1,a) & & & \\ (3,b) & (2,e) & & \\ (1,c) & (4,f) & (3,h) & \\ (5,d) & (2,g) & (5,i) & (4,j) \end{vmatrix}$$

Step 2. Generation of the set of T-subsequent edge T-matrices. For saving space, only edge T-matrices with non zero $|T''_m|_p$ will be listed below.

$$\underline{|T_2|} = \begin{vmatrix} (3,b) & & & \\ (1,c) & (3,h) & & \\ (5,d) & (5,i) & (4,j) & \\ . & (2,e) & (4,f) & (2,g) \end{vmatrix} \qquad \underline{|T_{2^2}|} = \begin{vmatrix} (1,c) & & & \\ (5,d) & (4,j) & & \\ . & (4,f) & (2,g) & \\ . & (3,h) & (5,i) & (4,j) \end{vmatrix}$$

$$\underline{|T_{24}|} = \begin{vmatrix} (3,b) & & & \\ (1,c) & (3,h) & & \\ . & (2,e) & (4,f) & \\ . & . & . & (2,g) \end{vmatrix} \qquad \underline{|T_{2^24}|} = \begin{vmatrix} (1,c) & & & \\ (5,d) & (4,j) & & \\ . & (3,h) & (5,i) & \\ . & . & . & (2,e) \end{vmatrix}$$

$$\underline{|T_{2^234}|} = \begin{vmatrix} (1,c) & & & \\ . & (3,h) & & \\ . & . & (5,i) & \\ . & . & (2,e) & (2,g) \end{vmatrix}$$

From $\underline{|T_1|}$, $|T_1|_p = 0$, since deleting the outgoing edge (1,a) from node 1, leads to $S_1^1 = 0$.

$$|T_2|_p = b(d+i) \qquad |T_2^2|_p = dg \qquad |T_{24}|_p = 0$$

$$|T_{2^24}|_p = dh \qquad |T_{2^234}|_p = hi.$$

Step 3. Evaluation of T".

$$T" = b(d+i) + dg + dh + hi.$$

Remark 5.10 Theorem 5.7 is intentionally derived for computing $|T"|$ from the set of T-subsequent edge T-matrices of the original graph, from which other k-tree summation of the graph can be also determined. If only $|T"|$ is of interest, it is simpler to calculate $|T"|$ by applying the algorithm given in Section 5.3.1 for evaluation of 1-tree summation.

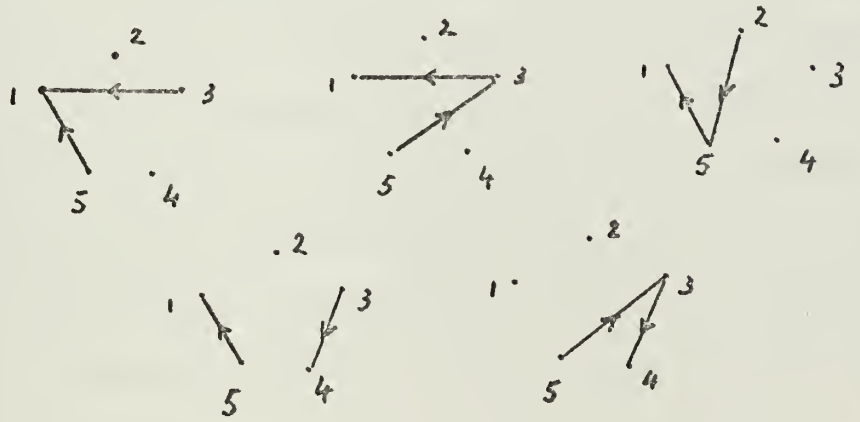


Figure 5.5.b. Illustration of Trees of $G_{14,23}$ Derived from the Graph of Figure 5.5.a.

VI. APPLICATION TO TOPOLOGICAL ANALYSIS OF ELECTRICAL ONE-PORT AND TWO-PORT NETWORKS

A. INTRODUCTION

In this chapter, algorithms developed in Chapter 5 will be proved to be useful for the determination of network functions of one-port and two-port networks at the same time with their sensitivity functions. Topological formula for computing network sensitivity function without actual derivation operation will be derived in Section 6.3. Application of the obtained formula requires computation of a k-tree admittance product summation of k-trees which do not contain a specified edge of the network. Technique for evaluating such a k-tree product summation will be given. Procedure for computation of network functions and its sensitivity function through application of the edged T-matrix will be discussed in Section 6.4. The merit of the process will be pointed out by the fact that all terms necessary for the computation of network functions and their sensitivity functions can be obtained simultaneously instead of using repeatedly several times an existing 1-tree finding program.

B. CHOICE OF TOPOLOGICAL FORMULAS

Topological formulas for analyzing passive one-port and two-port networks with no magnetic coupling have been derived by Percival [25], Maxwell [18], and Mayeda and Seshu [20]. Recently, using the concept of digraph,

Chen [10] has shown that lumped linear on-port and two-port networks including non reciprocal elements and mutual inductance can also be analyzed by topological formulas which are identical to those given for the case of passive networks, provided that directed tree summations are required instead of non directed tree summations. These formulas will be given below.

For a one-port network given in Figure 6.1, the driving point admittance is

$$Y_d = \frac{T}{T(11')} \tag{6.1}$$

where T and T(1,1') denote the admittance 1-tree summation and the Type-I 2-tree summation defined with respect to nodes 1 and 1' of network N, respectively.

It has been shown [20] that the driving point impedance Z_d at nodes 1 and 1' of network N can be expressed in terms of the impedance cotree summation of network N and network $N_{11'}$, derived from N by short-circuiting node 1 and 1'. For our purpose, it is better to express Z_d as shown in Eq. 6.2.

$$Z_d = Y_d^{-1} = \frac{T(1,1')}{T} \tag{6.2}$$

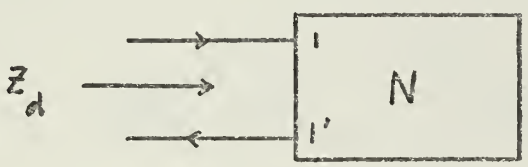


Figure 6.1. A General One-port Network.

For a general two-port network shown in Figure 6.2, the short-circuit admittance matrix is given by

$$Y_{sc} = \frac{1}{T''_r} \begin{vmatrix} T(2,3) & T(2,4;3,1) - T(2,1;3,4) \\ T(1,3;4,2) - T(1,2;4,3) & T(1,4) \end{vmatrix} \tag{6.3}$$

and the open circuit impedance matrix is given by

$$Z_{oc} = \frac{1}{T_r} \begin{vmatrix} T(1,4) & T(2,1;3,4) - T(2,1;3,4) \\ T(1,2;4,3) - T(1,3;4,2) & T(2,3) \end{vmatrix} . \tag{6.4}$$

where T_r denotes the summation of directed 1-tree admittance products with reference node r in the network N, and T''_r the summation of directed 1-tree admittance products with reference node r in the graph $N_{14,23}$ obtained from N by identifying nodes 1 and 4 then identifying nodes 2 and 3 as shown in Figure 6.3.

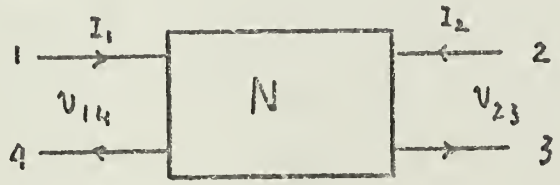


Figure 6.2. A General Two-port Network N.

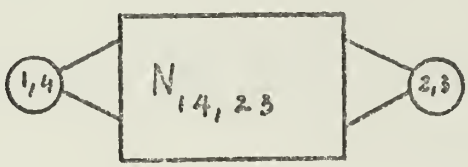


Figure 6.3. Network $N_{14,23}$ Derived from N by Identifying Nodes 1, 4 Then Identifying Nodes 2, 3.

From Eq. 6.3 and 6.4, the current ratio transfer function and voltage ration transfer can be determined as follows.

$$\alpha_{12} = \left.\frac{I_2}{I_1}\right|_{V_2=0} = \frac{T(1,3;4,2) - T(1,2;4,3)}{T(2,3)} \qquad (6.5.a)$$

$$\alpha_{21} = \left.\frac{I_1}{I_2}\right|_{V_1=0} = \frac{T(2,4;3,1) - T(2,1;3,4)}{T(1,4)} \qquad (6.5.b)$$

$$\mu_{12} = \left.\frac{V_2}{V_1}\right|_{I_2=0} = \frac{T(1,2;4,3) - T(1,3;4,2)}{T(1,4)} \qquad (6.6.a)$$

$$\mu_{21} = \left.\frac{V_1}{V_2}\right|_{I_1=0} = \frac{T(2,1;3,4) - T(2,4;3,1)}{T(2,3)} \qquad (6.6.b)$$

C. TOPOLOGICAL FORMULAS FOR DETERMINATION OF SENSITIVITY FUNCTIONS

The purpose of this section is to derive a topological formula for computing the sensitivity function of a network, when one of its parameters, say e_i, varies. To this end, let F be a driving point function of a one-port network or a transfer t function of a two-port network, the sensitivity of F to changes of the element e_i is defined by the following relation, as given by Goldstein and Kuo [13]:

$$S_{e_i} = \frac{\partial \ln F}{\partial \ln e_i} = \frac{e_i}{F} \frac{\partial F}{\partial e_i} \qquad (6.7)$$

1. The Bilinear Form of Network Functions

Inspecting Eqs. 6.1 to 6.6 shows that in general a network function can be represented as:

$$F = \frac{U}{V} \tag{6.8.a}$$

where U and V are Type-I 2-tree products summation or 1-tree products summation if F is either a driving point immittance function of a one-port network or a short-circuit admittance or an open-circuit impedance function of a two-port network, and U is the difference of two Type-III 2-tree products summation and V a Type-I 2-tree products summation. In all cases, it is possible to write:

$$F = \frac{y_i A + B}{y_i C + D} \tag{6.8.b}$$

where y_i denotes the admittance of an arbitrary element of the network. Eq. 6.8.b is obtained by partitioning U and V into two sums, of which one contains all tree products having y_i as a factor, and the other is form by all tree product that do not contain y_i. Eq. 6.8.b is called the bilinear form of the network functions.

Incidentally, it is interesting to note that the generalized Thevenin theorem can be obtained from Eq. 6.8.b. In this order, rearrange this equation as follows

$$F = \frac{y_i \frac{A}{C} + \frac{B}{D}\frac{D}{C}}{y_i + \frac{D}{C}} \tag{6.8.c}$$

Note that D and C can be considered as the 1-tree admittance product summation and a Type-I 2-tree admittance product summation defined with respect to the terminal of y_i and computed respectively from the network N', derived from N by identifying node 1 and 4. The ratio

$$W = \frac{D}{C} \tag{6.8.d}$$

is, by applying Eq. 6.1, the Thevenin admittance seen looking back into the network from terminals of y_i. Furthermore, $\frac{A}{C}$ and $\frac{B}{D}$ are clearly the limits of F when y_i is very large or zero, respectively. Therefore

$$F = \frac{Y_1 F_i + W F_o}{y_i + W} \tag{6.8.e}$$

Eq. 6.8.e is exactly the generalized Thevenin theorem as obtained by De Claris [11], Parker [23] and Sorensen [29].

2. Evaluation of Network Sensitivity Function

In this section a general formula for computing network sensitivity functions will be derived. To this end, taking the derivative with respect to y_i of Eq. 6.8.b, and substituting into Eq. 6.7 leads to

$$S_{y_i} = \frac{D}{y_i C + D} - \frac{B}{y_i A + B} \tag{6.9.a}$$

or

$$S_{y_i} = \frac{D}{V} - \frac{B}{U} \tag{6.9.b}$$

Suppose that the network function under consideration, F, has been evaluated, then U and V are known. Eq. 6.9.b shows that the sensitivity of F with respect to y_i can be obtained if D and B are also known. Since D and B can be derived from V and U respectively be setting $y_i = 0$. Then, D and B can be obtained from V and U respectively by eliminating all of the terms containing y_i. Therefore

Rule 6.1 Given a subsequent edge T-matrix $\lfloor T_m$. Let U_m and V_m be the partial summation of U and V computed from $\lfloor T_m$. The partial summations B_m and D_m computed from $\lfloor T_m$ are

(a) $B_m = U_m$ and $D_m = V_m$

if the entry representing $y_i \notin \lfloor T_m$.

(b) If the entry representing $y_i \in \lfloor T_m$, then set this entry equal to zero, and apply the same process for evaluate U_m and V_m for computing B_m and D_m.

It is clear that summing all B_m and D_m obtained by applying Rule 6.1 gives B and D.

D. COMPUTATION OF NETWORK FUNCTIONS AND NETWORK SENSITIVITY

The set of topological formulas listed above shows that the computation of network functions requires usually an 1-tree admittance products summation and/or a two-tree admittance product summations of Type-I and a difference of two 2-tree product summations of Type-III. Since the Type-I 2-tree summation T(a,b) can be obtained as an 1-tree summation from the graph derived from the original graph by coalescing nodes a and b, and since no algorithm for evaluating type-III 2-tree summation, T(a,b;r,c) has been available, network theorists have been directed to modify topological formulas for analyzing two-port networks in order to compute transfer functions by means of repeated application of a single 1-tree finding algorithm. To this end, Chan and Chan [5], have given a formula for evaluating

difference of two uncancelled type-III 2-tree summations, involved in the off diagonal elements of the Z_{oc} and Y_{sc} of Eqs. 6.3 and 6.4, into an algebraic sum of distinct type-I 2-tree summations. Farber and Malik transformed that difference into a difference of two uncancelling terms, but two intersection operations are required for the computation of these two terms [12]. Both of these two modified formulas are usually inefficient for large networks when a topological approach is seen at its best, because a large computer memory is required for the storage of all of the type-I 2-tree generated and in addition the intersection operation and the algebraic summation involved are time consuming. This difficulty can be circumvented by application of Theorem 5.6. Furthermore, no repeated application of a program is necessary, since all of the various k-tree summations required for the computation of a network function can be obtained from the same set of T-subsequent edge T-matrices of the original graph. This can be better illustrated by the following example. This example is identical to the one given in [5].

Example 6.1 Compute the short-circuit admittance matrix and the open circuit impedance matrix and their sensitivity, with respect to the admittance b_1, of the two-port network given in Figure 6.4.

Step 1. Form the augmented graph by adding fictitious edges, A and B, between nodes 1 and 4, and then between

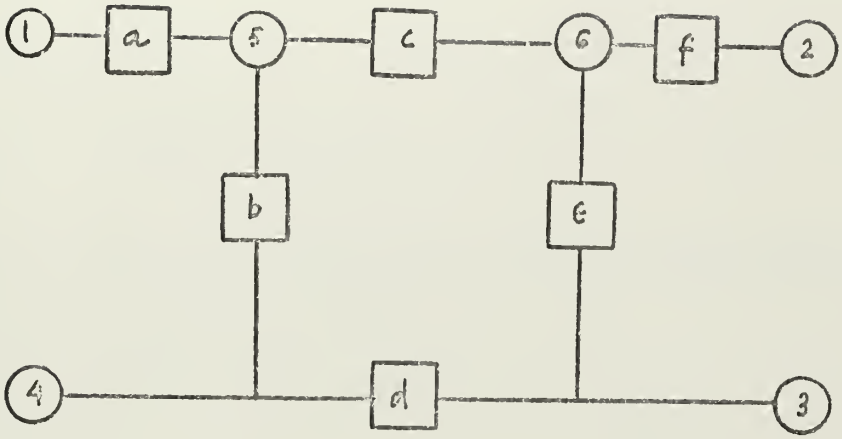


Figure 6.4. Two-port Network for Example 6.1.

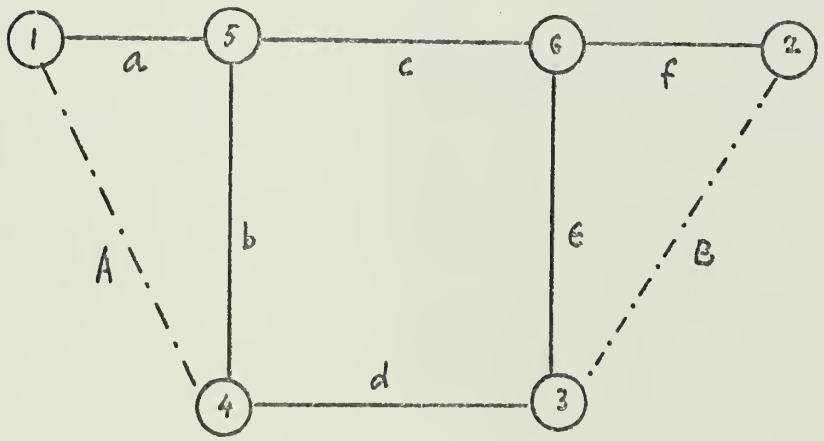


Figure 6.4.a. Augmented-Graph Associated with Network of Figure 6.4.

nodes 2 and 3, respectively. Associate with the augmented graph its edge T-matrix:

$$\underline{|T_1|} = \begin{array}{|cccccc|}\hline a & & & & & \\ . & c & & & & \\ . & . & e & & & \\ . & . & f & B & & \\ A & b & . & d & . & \\ \hline \end{array}$$

$$N_1 = 1\ 5\ 6\ 3\ 2\ 4$$

Step 2. Generate from $\underline{|T_1|}$ the set of T-subsequent edge T-matrices and their node vectors.

$$\underline{|T_{124}|} = \begin{array}{|cccccc|}\hline A & & & & & \\ . & b & & & & \\ . & . & c & & & \\ . & d & . & e & & \\ . & . & . & f & B & \\ \hline \end{array}$$

$1\ 4\ 5\ 6\ 3\ 2$

$$\underline{|T_{13^34}|} = \begin{array}{|cccccc|}\hline a & & & & & \\ A & b & & & & \\ . & . & d & & & \\ . & . & . & B & & \\ . & . & . & e & f & \\ \hline \end{array}$$

$1\ 5\ 4\ 3\ 2\ 6$

$$\underline{|T_{14}|} = \begin{array}{|cccccc|}\hline a & & & & & \\ . & c & & & & \\ . & . & f & & & \\ A & b & . & . & & \\ . & . & . & B & d & \\ \hline \end{array}$$

$1\ 5\ 6\ 2\ 4\ 3$

$$\underline{|T_{12^43^24^2}|} = \begin{array}{|cccccc|}\hline A & & & & & \\ . & d & & & & \\ . & . & e & & & \\ . & . & . & f & & \\ . & . & . & c & . & \\ \hline \end{array}$$

$1\ 4\ 3\ 6\ 2\ 5$

$$\underline{|T_{12^44}|} = \begin{array}{|cccccc|}\hline A & & & & & \\ . & b & & & & \\ . & d & . & & & \\ . & . & . & B & & \\ . & . & . & e & f & \\ \hline \end{array}$$

$1\ 4\ 5\ 3\ 2\ 6$

$$\underline{|T_{12^43^25}|} = \begin{array}{|cccccc|}\hline A & & & & & \\ . & d & & & & \\ . & . & B & & & \\ . & . & e & f & & \\ . & . & . & . & c & \\ \hline \end{array}$$

$1\ 4\ 3\ 6\ 5\ 2$

$$\underline{|T_{14^2}|} = \begin{array}{|cccccc|}\hline a & & & & & \\ . & c & & & & \\ A & b & . & & & \\ . & . & . & d & & \\ . & . & . & . & B & \\ \hline \end{array}$$

$1\ 5\ 6\ 4\ 3\ 2$

$$\underline{|T_{12^445}|} = \begin{array}{|cccccc|}\hline A & & & & & \\ . & b & & & & \\ . & d & . & & & \\ . & . & . & e & & \\ . & . & . & . & f & \\ \hline \end{array}$$

$1\ 4\ 5\ 3\ 6\ 2$

$$\left|\underline{T_{12^45}}\right| = \begin{vmatrix} A & & & & & \\ . & b & & & & \\ . & . & c & & & \\ . & . & . & f & & \\ . & . & . & . & B & \end{vmatrix} \qquad \left|\underline{T_{13^345}}\right| = \begin{vmatrix} a & & & & & \\ A & b & & & & \\ . & . & d & & & \\ . & . & . & e & & \\ . & . & . & . & f & \end{vmatrix}$$

$1\ 4\ 5\ 6\ 2\ 3 \qquad\qquad 1\ 5\ 4\ 3\ 6\ 2$

$$\left|\underline{T_{145}}\right| = \begin{vmatrix} a & & & & & \\ . & c & & & & \\ . & . & f & & & \\ . & . & . & B & & \\ . & . & . & . & d & \end{vmatrix}$$

$1\ 5\ 6\ 2\ 3\ 4$

A As soon as each $\left|\underline{T_m}\right|$ is generated, compute the partial k-tree summation as follows.

1. Localization of A and B

Use the node vector to localize:

a) row f associated with node 4,

b) column u associated with node 2

c) column x associated with node 3

If A has not been eliminated by the process of deriving $\left|\underline{T_m}\right|$, then $t_{f1} = A$. Similarly,

$$B = \begin{cases} t_{u+1,x} & \text{if} \quad u > x \\ t_{x+1,u} & \text{if} \quad u < x \end{cases}$$

If $t_{f1} = 0$, then:

$$\left|T(1,4)_m\right|_p = 0, \quad \left|T(1,2;3,4)_m\right|_p = \left|T(1,3;2,4)_m\right|_p = 0,$$

and

$$\left|T''_m\right|_p = 0.$$

If B has been eliminated from $\left|\underline{T_m}\right|$, then:

$|T(2,3)_m|_p = 0$ and $|T''_m|_p = 0$.

2. Computation of $|T_m|_p$

Let A and B equal to zero. Form s_i^m for $1 \leq i \leq n-1$. Form the product of all s_i^m obtained to compute $|T_m|_p$.

3. Computation of $|T(1,4)_m|_p$

Set $s_f^m = 1$. Compute $|T(1,4)_m|_p$ by forming the product of all of s_i^m obtained.

4. Computation of $|T(2,3)_m|_p$

If $u > x$, set s_{u+1}^m. If $u < x$, set $s_{x+1}^m = 1$. To compute $|T(2,3)_m|_p$, form the product of all of s_i^m obtained.

5. Computation of $|T''_m|_p$

Set $s_f^m = 1$. If $u > x$, set $s_{u+1}^m = 1$. If $u < x$, set $s_{x+1}^m = 1$. Form the product of all of s_i^m obtained to compute $|T''_m|_p$.

6. Computation of $|T(1,2,;3,4)_m|_p$ and $|T(1,3;2,4)_m|_p$

If u and x are both smaller than f, then:

$$|T(1,2;3,4)_m|_p = 0,$$
$$|T(1,3;2,4)_m|_p = 0.$$

If $f < u,x$ then $|T(1,2;3,4)_m|_p$ and $|T(1,3;2,4)_m|_p$ are given by Eq. 5.15.b and Eq. 5.15.c, respectively.

If $u < f < x$ or $x < f < u$, then $|T(1,2;3,4)_m|_p$ and $|T(1,3;2,4)_m|_p$ are given by Eqs. 5.17.c and 5.17.d.

7. To evaluate the sensitivity function with respect to variation of element e_i, specify terminal nodes of e_i,

search for the entry associated with e_i, using the node vector. If this entry is not zero, then set it equal to zero and repeat (2) to (6) for computing sub-partial k-tree summations, otherwise these sub-partial k-tree summations are equal to their corresponding partial k-tree summations, respectively.

Repeat the above process for each variable e_i.

Step 3. Compute separately every k-tree summation by summing all of its partial summation obtained in Step 2. Substituting into Eqs. 6.3 and 6.4 to determine the network functions. Substitute into Eq. 6.9.b to determine the sensitivity functions of every network function.

The results obtained in Steps 2 and 3 are listed in Tables 6.1, to 6.4.

Example 6.1 shows that to obtain a general algorithm for computing network functions and their sensitivity functions, through application of Theorems 5.3 to 5.7 given in Chapter 5, it is necessary to generate only a subset of T-subsequent edge T-matrices of the original graph and compute partial k-tree summations and their sub-partial k-tree summation from these edge T-matrices, using techniques described in the above mentioned theorems.

For directed graphs, techniques given in Example 6.1 can also be used, provided that:

1. Definition 3.1.a is used for representing the digraph,

3. In computation of partial k-tree summation, all

TABLE 6.1

PARTIAL K-TREE SUMMATIONS OBTAINED IN STEP 2 OF EXAMPLE 6.1

m	$\lvert T_m \rvert_p$	$\lvert T(1,4)_m \rvert_p$	$\lvert T(2,3)_m \rvert_p$	$\lvert T(12;34)_m \rvert_p$	$\lvert T(13;24_m \rvert_p$	$\lvert T''_m \rvert_p$
1	acef(b+d)	acef	ace(b+D)	0	0	ace
1^4	acfbd	acfd	acfb	acfd	0	acf
12^4	0	bc(d+e)f	0	0	0	bc(d+e)
13^34	0	0	abd(e+f)	0	0	ad(e+f)
$12^43^24^2$	0	defc	0	0	0	0
12^43^25	0	0	0	0	0	d(e+f)c
12^44	0	0	0	0	0	bd(e+f)
14^2	0	0	acbd	0	0	acd
12^445	0	bdef	0	0	0	0
12^45	0	0	0	0	0	bcf
13^345	abdef	adef	0	0	0	0
145	0	0	acfd	0	0	0

TABLE 6.2

TOTAL K-TREE SUMMATIONS

T	$= acef(b+d) + acfbd + abdef.$
$T(1,4)$	$= acef + acfd + bc(d+e)f + defc + bdef + adef.$
$T(2,3)$	$= ace(b+d) = acfb + abd(e+f) + acbd + acfd.$
$T(1,2;3,4)$	$= acfd.$
$T(1,3;2,4)$	$= 0$
T''	$= ace + acf + bc(d+e) + ad(e+f) + d(e+f)c + bd(e+f) + acd + bcf.$

The results obtained do agree with those given in [5].

TABLE 6.3

SUB-PARTIAL K-TREE SUMMATIONS

m	$\lvert T'_m \rvert_p$	$\lvert T'(1,4)_m \rvert_p$	$\lvert T'(2,3)_m \rvert_p$	$\lvert T'(12;34)_m \rvert_p$	$\lvert T'(13;24)_m \rvert_p$	$\lvert T''_m \rvert_p$
1	acefd	acef	aced	0	0	ace
1^4	0	acfd	0	acfd	0	acf
$1^3 3^4$	0	0	0	0	0	ad(e+f)
$1^2 4^4 3^2 4^2$	0	defc	0	0	0	0
$1^2 4^4 3^2 5$	0	0	0	0	0	d(e+f)c
$1^4 4^2$	0	0	0	0	0	acd
$1^3 3^3 45$	0	adef	0	0	0	0
145	0	0	acfd	0	0	0

TABLE 6.4

K-TREE SUB-SUMMATIONS USED FOR COMPUTING SENSITIVITY FUNCTIONS

T'	$= acefd$
$T'(1,4)$	$= acef + acfd + defc + adef$
$T'(2,3)$	$= aced + acfd$
$T'(1,2;3.4)$	$= acfd$
$T'(1,3;2,4)$	$= 0$
T''	$= ace + acf + ad(e+f) + d(e+f)c + acd$

outgoing directed edges from the designated reference nodes are deleted. In particular, for the computation of $|T''_m|_p$ nodes 2 and 3 should be identified. This operation has been described in Theorem 5.7.

VII. CONCLUSION AND SUGGESTION FOR FURTHER RESEARCH

The edge T-matrix has been defined and used as a new tool for network representation. The properties of the edge T-matrix have been discussed. Methods for the transformation of edge T-matrices have also been given. This led to algorithms for solving the following basic problems in subgraph enumerations:

1. determination of a set of fundamental circuits,
2. determination of all paths between a given pair of nodes,
3. determination of all circuits of a graph,
4. determination of all segs of a graph,
5. evaluation of 1-tree summation,
6. evaluation of various 2-tree summations defined with respect to a set of 2, 3, and 4 nodes of a graph.

The case of directed graphs has also been covered. The proposed methods enable one to analyze a network with reduced memory requirement and computer time, as compared to conventional methods. Application of algorithms for the evaluation of 1-tree summation and various 2-tree summations in topological analysis of one-port and two-port networks proves to be useful, specifically in the determination of network functions and the calculation of network sensitivity functions.

The extension of the present method to the enumeration of general k-tree summation seems natural. Determination

of planar graphs, detection of isomorphism of two graphs, and the synthesis of communication nets with specified paths or switching functions through the edge T-matrix representation should be interesting.

APPENDIX A

DEFINITIONS OF SOME FAMILIAR TOPOLOGICAL TERMS

Definition A.1 Linear graph and its edges and nodes. A linear graph, or simply a graph, is a set of line segments, called edges and points called nodes or vertices, which are the end-points, or terminals, of the edges, interconnected in such a way that the edges are connected only, or incident at the vertices.

Definition A.2 Subgraph. A portion G_i containing a subset of the edges and nodes of a given graph is called a sub-graph of G.

Definition A.3 Connected and unconnected graph. A graph is said to be a connected graph, if there is at least one path between every pair of its nodes; otherwise the graph is an unconnected graph.

Definition A.4 Degree of a node. The degree of a vertex of a graph is the number of edges incident with that vertex.

Definition A.5 A graph is oriented or directed if in each edge there is an orientation. If no orientations are assigned to its edges, the graph is a non-oriented graph.

Definition A.6 Outgoing edge. An oriented edge is called an outgoing edge with respect to a node if it is directed away from that node.

BIBLIOGRAPHY

1. Belevitch, V., "On the Realizability of Graph with Prescribed Circuit Matrices," in Switching Theory in Space Technology, pp. 126-144, Stanford University Press, Stanford, California, 1963.

2. Branin, F. H., "The Relation Between Kron's Method and the Classical Methods of Network Analysis," IRE Wescon Conv. Rec., Part 2, pp. 3-28, 1959.

3. Bryant, P. R., "Graph Theory Applied to Electrical Networks," in Graph Theory and Theoretical Physics, Academic Press, London and New York, pp. 111-137, 1967.

4. Cartwright D. and Gleason, T. C., "The Number of Paths and Cycles in a Digraph," Psychometrika, Vol. 31, pp 179-199, June 1966.

5. Chan, S. P. and Chan, S. G., "Modification of Topological Formulas," IEEE Trans. on Circuit Theory, vol. CT-15, no. 1, pp. 84-86, March 1968.

6. Chang, W. T., "Determination of Network Functions by Topological Method," Thesis, Naval Postgraduate School, 1968.

7. Char, J. P., "Circuit, Out-set and Path Enumeration and Other Applications of Edge Numbering Convention," Proc. IEEE (London), vol. 117, no. 3, pp. 532-538, March 1970.

8. Chen, W. K., "Unified Theory on the Generation of Tree of a Graph. Part I, The Wang Algebra Formulation," Int. J. Electronics, vol. 27, no. 2, pp. 101,117, 1969.

9. Chen, W. K., "Unified Theory on the Generation of Tree of a Graph. Part II, The Matrix Formulation," Int. J. Electronics, vol. 27, no. 4, pp. 319-336, 1969.

10. Chen, W. K., "Modification of Topological Formulas," IEEE Proc., Vol. 57, no. 12, pp. 2166-2167, Dec. 1969.

11. De Claris, N., "Driving Point Impedance Functions of Active Networks," IRE Conv. Rec. Vol. 4, no. 2, Circuit Theory, pp. 26-37, 1956.

12. Farber, L. and Malik, N. R., "A New Modification of Topological Formulas," IEEE Trans. on Circuit Theory, vol. CT-16, no. 2, pp. 89-91, Feb. 1969.

13. Golstein, A., and Kuo, F., "Multiparameter Sensitivity," IEEE Trans. on Circuit Theory, vol. CT-8, pp. 172-178, May 1961.

14. Gotlieb, C. C. and Corneil, D. G., "Algorithm for Finding a Fundamental set of Cycles for an Undirected Linear Graph," Com. of the ACM, vol. 10, no. 12, pp. 780-783, Dec. 1967.

15. Kamea, T., "A Systematic Method of Finding all Directed Circuits and Enumeration All Directed Paths," IEEE Trans. on Circuit Theory, vol. CT-14, pp. 166-171, June 1967.

16. Kirchhoff, G., "Uber die auflosung der gleichungen, auf welche man bei der untersuchung der linear vertheilung galvanischer strome gefuhrt wird," Annalen der Physik und Chemie, vol. 72, pp. 497-508, 1847; English translation, IRE Trans. on Circuit Theory, vol. CT-5, no. 1, pp. 4-7, March 1958.

17. MacMahon, P. A., "Combinatory Analysis," vol. 1, pp. 112-113, Cambridge University Press, 1915.

18. Maxwell, J. C., "Electricity and Magnetism," Oxford: Clarendon Press, 1892.

19. Maxwell, L. M. and Reed, G. B., "Subgraph Identification Seg, Circuits and Paths," Presented at the 8th Midwest Symposium on Circuit Theory, 1965.

20. Mayeda, W. and Seshu, S., "Topological Formulas for Network Functions," Bulletin 446, University of Illinois Engineering Experiment Station, 1957.

21. Nakagawa, N., "On Evaluation of the Graph Tree and the Driving Point Admittance," IRE Trans. on Circuit Theory, vol. CT-15, pp. 122-127, June 1958.

22. Okada, S., "Topology Applied to Switching Circuits," Proc. Symp. on Information Network, Polytechnic Institute of Brooklyn, April 1954.

23. Parker, S. R., "The Effect of Parameter Variations Upon the Behavior of Electrical Networks," Sc. D. Dissertation, Stevens Institute of Technology, Hoboken, New Jersey, 1964.

24. Parthasarathy, K. R., "Enumeration of Paths in Digraphs," Psychometrica, vol. 29, pp. 152-165, June 1964.

25. Percival, W. S., "Solution of Passive Electrical Networks by Means of Mathematical Trees," Proc. IRE (London), vol. 100, pt. III, pp. 143-150, May 1953.

26. Riordan, J., "An Introduction to Combinatorial Analysis," pp. 5-6, John Wiley & Son Inc. 1958.

27. Rode, F. and Chan, S. P., "Computer Evaluation of Topological Formulas for Network Analysis," Proc. of the First Hawaii International Conference on System Sciences, pp. 790-802, Jan. 1968.

28. Seshu, S., and Reed, M. B., "Linear Graphsand Electrical Networks," Reading Mass.: Addison-Wesley Publishing Company, Inc. 1961.

29. Sorensen, E. V., "General Relation Governing the Exact Sensitivity of Linear Networks," IEEE (London), Proc. vol. 114, no. 9, Sept. 1967.

30. Veblen, O., "Analysis Situs," vol. 5, Pt. 2, American Math. Soc., Cambridge Colloquium Publication, 1922.

31. Veblen, O., and Franklin, P., "On the Matrices Whose Elements are Integers," Annals. Math., vol. 23, pp. 1-15.

32. Welch, J. T.,Jr. "A Mechanical Analysis of the Cyclic Structure of undirected Linear Graphs," J. ACM, vol. 13, no. 2, pp. 205-210, April 1966.

33. Wing, O., and Kim, W. H., "The Path Matrix and its Realizability," IRE Trans. on Circuit Theory, vol. CT-6, pp. 267-272, Sept. 1959.

INITIAL DISTRIBUTION LIST

		NO. Copies.
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Assoc. Professor S. G. Chan Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Miss. Le Dau Thu 118 Dong Khanh Cholon Viet-Nam	1

Blank

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

EDGE T-MATRIX IN NETWORK THEORY

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Electrical Engineer's Thesis; September 1970

5. AUTHOR(S) (First name, middle initial, last name)

Le Phung, Lieutenant, Viet-Nam Navy

6. REPORT DATE

September 1970

7a. TOTAL NO. OF PAGES

149

7b. NO. OF REFS

33

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

Theory of the edge T-matrix will be developed and applied to derive algorithms for solving basic problems of network theory such as the determination of a fundamental loop or cut-set matrix, the path matrix, the circuit matrix, the seg matrix and the tree summation calculation. Application of the tree summation to the determination of sensitivity functions without actual derivative operation will also be investigated. Formulas for determining topologically the sensitivity function will be proved.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
GRAPH						
NETWORK						
SUBGRAPH						
CIRCUIT						
PATH						
SEG						
FUNDAMENTAL LOOP						
FUNDAMENTAL SEG						
TREE						
K-TREE						
ALGORITHM						
CIRCUIT ANALYSIS						
SENSITIVITY						
TOPOLOGY